TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY
Julkaisu 547 • Publication 547

Anneli Heimbürger

It's Time to Link!
On the Development of Time-Sensitive Linking Structures for the Web

Tampere 2005

Tampereen teknillinen yliopisto. Julkaisu 547
Tampere University of Technology. Publication 547

Anneli Heimbürger

It's Time to Link!
On the Development of Time-Sensitive Linking Structures for the Web

Thesis for the degree of Doctor of Technology to be presented with due permission for public examination and criticism in Auditorium 125, at Tampere University of Technology - Pori, on the 21st of October 2005, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2005

ISBN 952-15-1448-5 (printed)
ISBN 952-15-1483-3 (PDF)
ISSN 1459-2045

Heimbürger, Anneli
It's Time to Link!
On the Development of Time-Sensitive Linking Structures for the Web
Tampere University of Technology, Publication 547, Tampere 2005-09-14

Keywords: XML Linking Language, XLink, context, time-sensitive linking, professional Web applications

Abstract

Context dependent access is becoming the most relevant means of retrieving information from both the traditional and mobile Web. The user, location, device and time can be regarded as dimensions of the context. For example, in industrial maintenance situations a maintenance crew onsite needs step-by-step instructions for their devices. Time is a fundamental parameter because the maintenance procedure descriptions can depend on the service intervals of the technical object. For example, if time $t_1 < t \leq t_2$ then document group $D_{g1} = (d_1, d_2, d_3)$ will be delivered to the user. If $t_2 < t < t_3$, then the deliverable document assembly may be $D_{g2} = (d_1, d_4, d_5)$. The problem is how to connect the right document groups with associated time contexts. Other examples of time-sensitive Web applications are time-based project management and communication in distributed environments, time-sensitive filters for Web-based news services, composing and presenting training resources from a time-based point of view.

In this thesis an XLinkTime model for time-sensitive linking structures is introduced. It is demonstrated in this thesis that XLink's (XML Linking Language) multiple linking, link traversal rules and linkbases with a timerule namespace extension can be used to realize time-sensitive linking structures in professional Web-based applications. Proof-of-concept implementations have been carried out in two application domains: project management and usage of technical manuals.

Preface

> *"The Semantic Web is still a web, a collection of linked nodes. Navigation of links is currently, and will remain for humans if not machines, a key mechanism for exploring the space. The Semantic Web is viewed by many as a knowledge base, a database or an indexed and searchable document collection, ... we view it as hypertext." (Bechhofer, Carr, Goble and Hall 2001)*

Be the Web semantic or something else in the future, the simple fact is that contents and links are the core of it. Context-based linking and information retrieval in all areas of knowledge work with all possible user platforms will be the essential way of accessing information from the Web.

I was thinking of a choice between two possibilities when I settled on the title for my thesis. The first option was a purely academic title and the second option is now the title of this thesis. The first sentence, "It's time to link!", is symbolic and includes the two main objects of this thesis, "hypertext links" and "time", along with the dynamics between them. It is also an expression of concern and hope that future development of linking on the Web will converge towards solidly based, more versatile solutions. The second sentence of the thesis title, "On the Development of Time-Sensitive Linking Structures for the Web", is more academic and describes the content of this thesis.

My research interests and therefore approach in this thesis are more practical and application domains oriented than theoretical and formalism oriented. Hypertext and hypermedia link management issues form the greater background framework for my thesis. Link management is meant to be the essential glue of this thesis, binding the parts together and providing a perspective view of links from the past to the future. I have worked in the field of hypermedia for almost 20 years, mostly with applied research projects. I have had the opportunity to follow how hypermedia research and link management have progressed during the past 20 years, going from stand-alone systems to distributed world wide hypertext, as well as how links have been treated in this process. In stand-alone hypertext environments we had rich linking possibilities, but we have lost most of them in distributed, world-wide environments. Now, however, more versatile linking functions can be realized in the Web, at least in professional applications.

I use Harvard system for references in my thesis, because I prefer to keep the semantics provided by authors' and researchers' names in the text instead of pure numbers according to Vancouver system. With figures, the reference to an original source is given in the text at the end of the sentence in which the figure is referred.

This thesis has deep roots. Its main idea was discussed already in 1998 as visionary thought in the context of industrial trouble-shooting situations. An example is the onsite maintenance crew needing step-by-step instructions quickly for their devices. Time is a fundamental parameter because maintenance procedure descriptions may depend on how long the situation has continued. The focus of my research approach is on how time-

sensitive linking structures can be modeled and realized. One possibility is to extend XML Linking Language (XLink), which is the recommendation defined by the World-Wide Web Consortium (W3C). I am sure that there could be other approaches too. My approach is based on linking.

This work has mainly been carried out in the Advanced Multimedia Center (AMC) at the Tampere University of Technology (TUT), Pori, during the years 2003 – 2005.

I would like to thank Professor Hannu Jaakkola, the director of Pori Doctoral School, for his valuable views which have helped me to choose the right way in my life and also for always being virtually present during this thesis process. I want to express my great gratitude to the head of the Advanced Multimedia Center, Professor Jari Multisilta, for providing me with the opportunity to work in his research group, full of challenges. I also want to thank Professor Multisilta for acting as my supervisor, and patiently listening to or reading my sometimes more and sometimes less intelligent thoughts in emails. His professional knowledge, encouragement and constructive feedback for my work have been indispensable. This dissertation process would not have begun, continued, and finally finished without him. I want to thank Senior Researcher Jari Palomäki for interesting, supportive and encouraging discussions on academic traditions and the philosophical issues of science.

I wish to thank the AMC personnel for providing a working environment full of understanding, joy and nice surprises. I especially wish to thank Research Assistant Kai Ojansuu for his in-depth know-how in Java technology. My special thanks go to the technical staff of Pori for their kind help during my research work, the staff of the Pori Science Library for their flexible services, to Mrs. Eija Kentta for her help in project management issues and to Mrs. Ulla Nevanranta for her assistance with regard to European-Japanese Conference practices.

I would like to thank Professor Jaak Henno, of Tallinn Technical University, Estonia and Professor Pasi Tyrväinen, at the University of Jyväskylä, Finland for their valuable and insightful comments and remarks for improving the introductory part of this thesis.

Dr. Ted Nelson, Fellow of the Oxford Internet Institute and Visiting Fellow of Wadham College, University of Oxford, inspired me to the world of hypertext already in 1988. In 1989 I had an opportunity to organize his visit to Finland. We held the Hypermedia Seminar 5.9.1989 at VTT. He is still online.

For revisions of the English manuscript I am grateful to Mrs. Sylvia Saila-Kuusman. Lecturer Marketta Rostedt provided invaluable help in revising and improving my English language in some critical papers. The remaining errors are mine. I thank Ms. Eva Brander for helping me with the bibliography and with some figures in this thesis.

My research work received funding from the following sources: the National Technology Agency of Finland (Tekes), the Finnish Cultural Foundation (Satakunnan maakuntarahasto), the High Technology Foundation of Satakunta (Satakunnan korkean teknologian säätiö) and the Foundation of Ulla Tuominen (Ulla Tuomisen säätiö). I wish to acknowledge their support. I would also like to take this opportunity to thank Jouko Hautamäki (Tekes), Ilkka Palola (Citec Information Inc.), Päivi Leppänen (Innovative Ideas Inc.), Timo Kunnas (Kronodoc Inc.) and Marko Mikkola (Prizztech Inc.) for co-operation within the XLinkIT research project related to this thesis.

I would like to express my gratitude to my father in eternity and my mother, fortunately still with us, for their upbringing and encouragement to face the challenges of life. My dear sisters Doris, Marianne and Rea, and my dear brother Harri; I am so lucky to have you all there. Dear friends, the researcher's soul of mine thanks you for your deep understanding that this kind of work is a way of life for me.

In a world where everyone seems to be larger and louder than me, it is very comforting to have a small quiet friend, in my case many of them. I thank all my teddy bears for their silent and understanding companionship. Special warm thanks go to Hunaja Mussukka for his wise, faithful support.

There will also be life after this thesis. I would like to dedicate this work to the promotion of the scientific research and development work between Finnish and Japanese universities.

Pori 14.9.2005

Anneli Heimbürger

Publications

Paper 1. Heimbürger, A. 1999. A structured link document as a new means for composing and publishing technical customer documentation in extranets and intranets. In: Smith, J. W. T, Ardö, A. and Linde, P. (Eds.) Proceedings of the 3rd ICCC/IFIP Conference on Electronic Publishing, 10 - 12 May 1999, Ronneby, Sweden. Washington, DC: ICCC Press. Pp. 90 - 102.

Paper 2. Heimbürger, A. 1999. Link management in technical customer documentation. In: Käkölä, T. (Ed.) Proceedings of the 22nd Information Systems Research Seminar in Scandinavia (IRIS 22), Enterprise Architectures for Virtual Organisations, 7 – 10 August, 1999, Keuruu, Finland. University of Jyväskylä, Department of Computer Science and Information Systems, Technical Reports TR-21, Vol. 2. Pp. 13 - 26.

Paper 3. Heimbürger, A. 2003. Modelling time-sensitive linking mechanisms. In: Jaakkola, H., Kangassalo, H, Kawaguchi, E. and Thalheim, B. (Eds.) Frontiers in Artificial Intelligence and Applications, Vol. 94, Information Modelling and Knowledge Bases XIV. Amsterdam: IOS Press. Pp. 26 - 42.

Paper 4. Heimbürger, A. 2005. Developing time-sensitive hypertext linking and navigation support. In: Nicholson, P., Thompson, J. B., Ruohonen, M. and Multisilta, J. (Eds.) E-Training Practices for Professional Organizations. Boston: Kluwer Academic Publishers. Pp. 291 – 298.

Paper 5. Heimbürger, A. 2003. Contextual multi-ended linking: Developing a method for content adaptation. In: Auer, M. and Auer, U. (Eds.) Proceedings of the International Workshop on Interactive Computer Aided Learning (ICL 2003), Learning Objects and Reusability of Content, 24 – 26 September, 2003, Villach, Austria. Pp. 1 - 25.

Paper 6. Heimbürger, A. 2005. Time-sensitive relationship management in technical manuals. Case: Maintenance schedules. In: Kiyoki, Y., Wangler, B., Jaakkola, H. and Kangassalo, H. (Eds.) Frontiers in Artificial Intelligence and Applications, Vol. 121, Information Modelling and Knowledge Bases XVI. Amsterdam: IOS Press. Pp. 152 – 169.

Table of contents

Abbreviations

ACM	Association for Computing Machinery
AHM	Amsterdam Hypermedia Model
CWI	Centrum voor Wiskunde en Informatica
DDL	Description Definition Language
DLS	Distributed Link Service
DPMC	Distributed Project Management and Communication
FOHM	Fundamental Open Hypermedia Model
HAM	Hypertext Abstract Machine
HBMS	Hyperbase Management Systems
HOSS	Hypermedia Operating System
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HyTime	Hypermedia/Time-based Structuring Language
ICT	Information and Communication Technology
ISO	International Standards Organisation
LSS	Link Server Systems
MESH	Maintainable, End User Friendly and Structured Hypermedia Model
MHEG	Multimedia and Hypermedia Information Coding Expert Group
MPEG	Moving Picture Experts Group
OHIF	Open Hypermedia Interchange Format
OHS	Open Hypermedia Systems
OHP	Open Hypermedia Protocol
OHSWG	Open Hypermedia Systems Working Group
RDF	Resource Description Framework
RNA	Relationship Navigation Analysis
RUP	Rational Unified Process
SGML	Standard Generalized Markup Language
SMIL	Synchronized Multimedia Integration Language
SVG	Scalable Vector Graphics
TEI	Text Encoding and Interchange
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium
WWW	World Wide Web
XHTML	Extensible Hypertext Markup Language
XLink	XML Linking Language
XLinkTime	Time-sensitive Linking Structure
XML	Extensible Markup Language
XPointer	XML Pointer Language
XSLT	Extensible Stylesheet Language Transformations
XTM	XML Topics Maps
XUL	XML User Interface Language

List of Figures

List of Tables

Section 1

Introduction

Hypermedia integrates digital computing with the natural tendency of the human mind to think associatively. This associative structure of the mind is very different from the linear way in which information is generally organized. For example, a technical manual or a project management guidebook, as linear arrangements, represents a single way through a topic. But for human minds there are many possible ways and contexts through associative information structures. The fundamental technique of hypermedia – creating associative links between resources – shows great potential for knowledge work.

Knowledge work is a challenge for today's businesses, and it affects the position of enterprises competing internationally. Formal and informal knowledge is becoming dynamic and reusable within different business applications and processes. A systematic approach to creating, composing, delivering, maintaining and using knowledge and related documents in different application domains, such as in project management and in industrial engineering, is an issue of vital importance in enterprises. The challenge lies in keeping track of documents, getting the right resources and sub-resources for users, and in ensuring that information is presented in the right contexts. Providing right pieces of information on the right devices for the right users, in the right places at the right time is economically critical, for example, in monitoring project progress or in machine maintenance situations.

Web-based knowledge work, if it is to be efficient and economic in the long term at least, should be based on international standards set by the International Standards Organisation (ISO) and on recommendations by the World Wide Web Consortium (W3C). Common standards and recommendations provide a basis for consistent working methods in inter- and intra-organizational environments. Automatic content and content descriptive metadata management, together with automatic management of associated links combine to create a long term vision for systematic knowledge work. When context dependent applications, i.e. deliverable contents dependent on specific situations, are to be developed, hypermedia link management is one of the key issues to be considered. In context dependent linking applications content related links are activated in certain situations depending on the context parameters, i.e. a certain rule or a set of rules is valid.

Hypermedia technology developed both by the hypermedia research community and by the W3C community, has the potential of contributing to several areas of computer-based human work, from distributed project management to e-learning, from creative writing to technical documentation in the engineering industry.

In this thesis the text goes in three different, but tightly interlinked, levels. The main research work of this thesis, (1) developing time-sensitive linking structures for the Web, interlaces (2) ongoing discussions between the hypermedia research community and the World Wide Web Consortium. The apprehension of (3) link management on the Web is always present, although sometimes on stage and sometimes behind the scenes. The core message can be regarded as serious concern, but also as enthusiasm, for taking part in future development work at the W3C.

1.1 The Research Problem

The main goal of the dissertation is to develop time-sensitive linking structures for professional Web-based applications. Time-sensitive links are functions of time. They can be activated by a user when a certain temporal rule or a set of temporal rules are valid.

<u>The Research Problem of the thesis is:</u>

How do we construct time-sensitive linking structures for professional Web-based applications which support the connection of the right resources and/or portions of resources according to associated time contexts?

An example of time-sensitive contexts in industry is the maintenance operations of a technical machine which follows a certain maintenance schedule with defined service intervals. At a certain point in time or during a certain time interval, the maintenance crew onsite needs step-by-step instructions for their devices. Time is a fundamental parameter because the maintenance procedure instructions depend on service intervals. For example, if time $t_1 < t \leq t_2$ then the document group $D_{g1} = (d_1, d_2, d_3)$ will be delivered to the user. If $t_2 < t < t_3$ then the deliverable document assembly may be $D_{g2} = (d_1, d_4, d_5)$.

<u>The research problem can be divided into the following sub-problems:</u>

1. How can temporal relations be expressed?
2. What does time-sensitivity require from linking structures?
3. What are the possibilities of existing Web-based linking languages with W3C Recommendation status to support time-sensitive linking structures?
4. How can proof-of-concept implementations of time-sensitive linking structures be realized?

<u>The premises of this thesis have been derived from the observation phase of this research (Table 1) and can be summarized as follows:</u>

1. The most relevant time contexts and related resources and/or portions of resources can be identified and specified in the application design phase.
2. The amount of resources and the number of associated links are limited.
3. Time contexts are task and situation dependent.
4. A user's information needs are task and situation oriented as well as navigation directed.

5. The use environment for professional applications is the World Wide Web. XML (Extensible Markup Language) document repositories correspond to resources and/or portions of resources.
6. The realization of the solution should be based on W3C Recommendations because common standards and recommendations provide a basis for consistent working methods in inter- and intra-organizational environments.

1.2 Research Method and Design Technique

The research method and the design technique used in the thesis are described in this section with references to core articles. The constructive research method provides four main avenues of approaching a research problem in information system research: theory building, experimentation, observation and systems development (Nunamaker et al. 1991). Relationship Navigation Analysis (RNA) provides a systematic way of identifying useful relationships in application domains (Yoo and Bieber 2001), and is used in the thesis as a design technique. RNA is based on generic relationship taxonomy. Application developers can implement each relationship as a link.

1.2.1 The Constructive Research Method

Research can be divided into basic and applied research. Basic research tries to find out what is part of reality. In applied research the results of basic research are applied to a certain object in order to archive the desired final state. The focus of this thesis is on applied research. Basically, constructive research is applied research, but instead of a final product, it is possible to accept a prototype or even a plan (Järvinen 2001). However, sometimes the constructive research method can be interpreted as being applied to more theoretical construction of models. It is typical for constructive research to build new innovations and this process is based on existing knowledge and/or technical, organizational etc. advancements. The utility of a new innovation is sooner or later evaluated. Constructive research is also called design-science research when the study emphasizes both construction and improvement with an integrated outcome, called a technological rule (Järvinen 2004; Van Aken 2004).

The constructive research method, based on Nunamaker's multi-methodological approach to information systems research, (Nunamaker et. al 1991) has been used in this thesis. The multi-methodological approach consists of four research strategies: observation, theory building, systems development and experimentation. These strategies can include different research methods and they can also be the main phases of the research. According to Nunamaker, research can utilize any combinations and order of the four strategies. The research can start from any of these and proceed to any other, so the method defines no fixed order or number of phases to be used in the research. The usability of the method is based on its flexibility, iterative and problem oriented approach.

The research phases of this thesis are presented in Table 1. Several iterations between theory building, systems development and experimentation phases were carried out during the whole life cycle of the research.

Table 1. Research phases of the thesis

Research Strategy	Research Tasks
Observation	Examples of time-sensitive situations related to the two application domains of the thesis, distributed project management and technical documentation, were identified by means of discussions with domain experts and systems developers. The main practical premises for this thesis were identified and derived.
Theory building	An XLinkTime model for time-sensitive linking structures and an example of its realization were created.
Systems development	Temporal relationship analyses were carried out and alternatives for proof-of-concept implementations for time-sensitive linking structures were identified. Demonstrations were designed and implemented.
Experimentation	Demonstrations were used as a proof-of-concept to show the feasibility of the chosen solutions in theory building and systems development. Demonstrations were validated and evaluated in the laboratory.
Observation	Possibilities and shortcomings of the approach developed and issues for further research work were identified by discussions with experts and systems developers in the application domains.

1.2.2 Relationship Navigation Analysis

The core design of any hypermedia system is how it models relationships as links (Fielding et al. 1998). A vital aspect in the design process is to identify relationships and implement them as links. Few designers explicitly think about the interrelationships of their applications and whether users should access and navigate them directly. This is partly because few systems demonstrate rich link structures, and partly because few tools exist that would help designers to think about an application in terms of its relationships. Yoo and Bieber (2001) introduced RNA. RNA provides a systematic way of identifying useful relationships in application domains and implementing them as links (Balasubramanian et al. 2001). RNA enhances the understanding of application domains of system developers by broadening and deepening their conceptual model of the domain.

RNA is based on generic relationship taxonomy and it can be applied to describe an application domain in terms of its relationships. It provides a systematic technique for determining the relationship structure of an application, helping the designers to discover all potentially useful relationships in an application. Relationship Navigation Analysis consists of four main steps: stakeholder analysis, element analysis, relationship analysis and implementation feasibility analysis (Yoo and Bieder 2001). In the first phase those user groups who need time-sensitive aspects are identified. The second phase, element analysis, identifies different elements or element clusters of interest for each user group. In the third phase, time-sensitive relationships for each element or element cluster are analysed. Navigation analysis identifies possible navigational structures and implementation analysis evaluates existing technological methods and tools. Finally, designers and users can do an informal cost/benefit analysis based on iterative testing to decide which relationships would be useful and feasible enough to be included in the final version of the application.

According to Yoo and Bieber (2001) the third phase, relationship analysis, can be conducted at three levels of details: generic relationship taxonomy, domain independent categories and domain dependent categories. At the general level, an analyst uses broad generic relationship categories to examine each item of interest. RNA's generic relationship taxonomy contains 16 relationships (Table 2) (Yoo and Bieber 2001).

Table 2. RNA's generic relationship taxonomy

Relationships	**Connections**
Generalization relationship	Connects an item of interest to the items whose concepts include its concept in a taxonomy
Characteristic relationship	Connects an item of interest to its attributes, parameters, metadata and other background information
Descriptive relationship	Connects an item of interest to definitions, illustrations, explanations and other information
Occurrence relationship	Connects multiple instance/views/uses/transformations of the same object in different parts of a system
Configuration/Aggregation relationship	Connects a part to other parts or a whole functionally or structurally
Membership/Grouping relationship	Connects a member of collection to other members or a whole collection
Classification relationship	Connects an item of interest to its instance or class
Equivalence relationship	Connects instances of the exact same object to a given item, i.e. same copies of a book or an exact match in information retrieval
Similar/Dissimilar relationship	Connects all items that share some positive or negative degree of similarity
Ordering relationship	Puts items in some kind of sequence
Activity relationship	Deals with relationships that exist among elements that are involved in some kind of activity i.e. among an input, tool, and an output
Influence relationship	Connects an item of interest to the item over which it has some kind of influence i.e. causal or control relationship
Intentional relationship	Connects an item of interest to the goals, arguments, issues, decisions, opinions and comments associated with the item
Socio-organisational relationship	Connects an item of interest to the position, authority, alliance, role and communication associated with the item in a social setting or organisational structure
Temporal relationship	Connects an item of interest to temporally related items
Spatial relationship	Connects an item of interest to related items in spatial dimensions

The whole repertoire of the RNA taxonomy is not used in this research, only the main idea of RNA is applied (Paper 6). The four main steps of RNA, as described earlier, are utilised in the proof-of-concept implementation phase of the thesis. A temporal relationship analysis must be done in order to identify the time-sensitive linking structures of the application concerned.

1.3 Theoretical Background

The major work of the hypermedia community has been in defining hypermedia. The Dexter model (Halasz and Schwartz 1994) is the most essential result of their work. The original Dexter Hypertext Reference Model (Halasz and Schwartz 1994; Grønbæk and Trigg 1994) was the result of a series of workshops held between October 1988 and July 1990. The Dexter reference model tries to integrate and formalize aspects found in different hypertext systems. Among them are such well-known systems as Intermedia, KMS, NoteCards, and Augment. Intermedia represented the first step toward openness, although both link webs and resources had to live within the Intermedia environment. This weakness was pointed out by one of the developers, Norman Meyrowitz, in his famous "The Missing Link" article (1989). The link set idea has had a lasting influence on both the Dexter model and on later open hypermedia systems (Grønbæk and Trigg 1999). The Dexter model includes multiple linking. On the Web HTML (Hypertext Markup Language) links are uni-directional and single-headed (W3C 2004b).

The core idea in the Dexter model is that the various objects which are involved in representing hypermedia structures should be specializations of a single generic entity. This entity models the resources, the links used to interconnect them, and the composites used to represent hypermedia structures. The basic architecture of the Dexter model separates contents and links (Figure 1) (Halasz and Schwartz 1994). The Dexter group introduced a component to express a concept that unified the ideas of the link and the node (Figure 2, Figure 3) (Grønbæk and Trigg 1999).

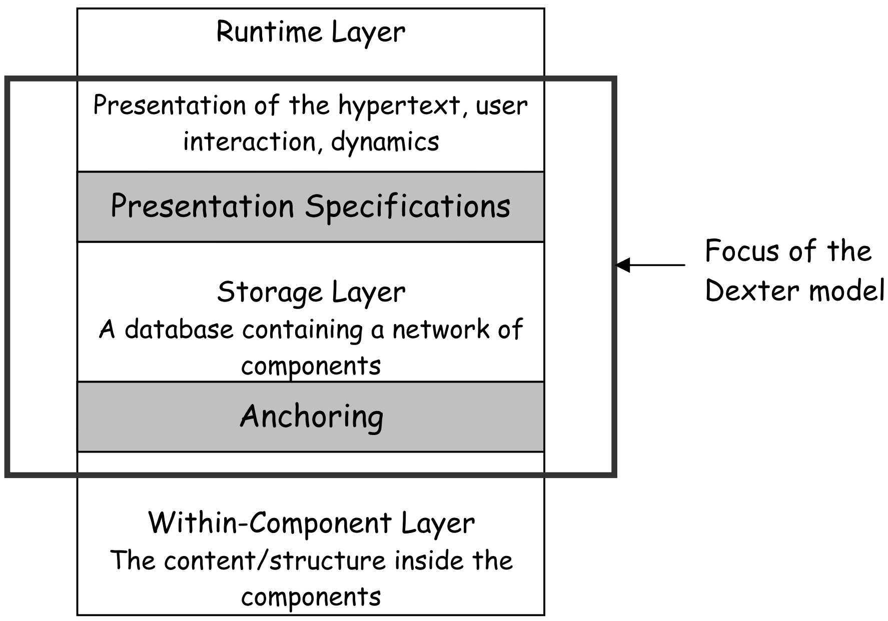


Figure 1. Dexter's three-layer architecture

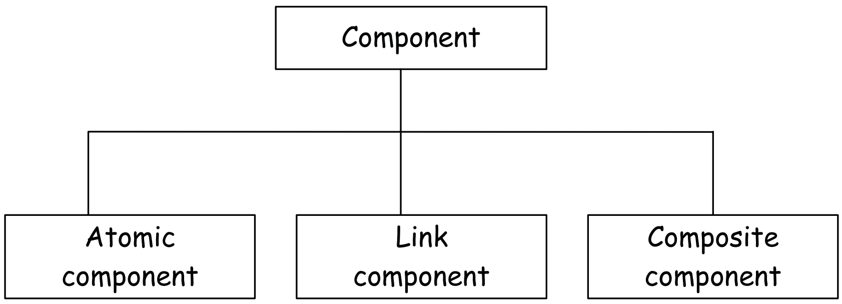


Figure 2. The Dexter model includes three basic components: an atomic component, a link component and a composite component

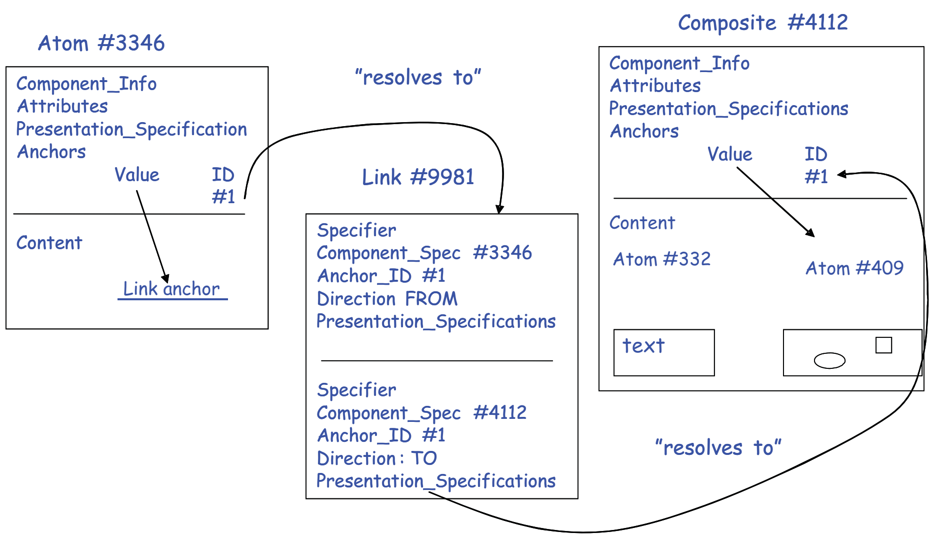


Figure 3. An example of an atom, a link and a composite

The basic principles of the Dexter model have been and still are the core for open hypermedia research directions. The Dexter model has affected design requirements and the development of XML Linking Language (XLink) (W3C 2001b). The model provides a theoretical background for this thesis. In addition to a theoretical background, it is important to understand the application domains under consideration. In this case the application domains are distributed project management and usage of technical manuals.

1.4 Basic Concepts

The definitions of the basic concepts used in this thesis are provided next.

The nuclear elements of *hypertext* are nodes and links (Berk and Devlin 1991). The fundamental unit of information in a hypertext is *a node*. A node may be as small as a word or as large as a book. Nodes are connected to each other by *links*. A link has two ends, called *anchors*, and a direction. The link starts at the source anchor and points to the destination anchor. To activate a link a user points to a source anchor, which is an on-screen indicator of the presence of a link. Hypertext contains textual information and hypermedia contains other forms of data also.

On the Web, *a resource* or *a portion of a resource* is any addressable unit of information. On the Web, the means used for *addressing* a resource or a portion of a resource is a *URI* (Uniform Resource Identifier) reference.

In an HTML-link, the text string between <a> and elements defines the source anchor. The *href*-attribute in each source anchor specifies the address of the destination anchor with an URI address. The source anchor may be a resource or a portion of a resource. The destination anchor may be any Web resource such as an image, a video clip, a sound sequence, a program, an HTML document, or a location within an HTML document. If the destination anchor is a location within an HTML document it must be given an *anchor name* and any URI addressing this anchor must include the name as its portion identifier.

XLink specification (W3C 2001b) defines the *XML Linking Language*, which allows elements to be inserted into XML documents in order to create and describe links between resources. In XLink, a link between resources or portions of resources is made explicit by an *XLink linking element*, which is an XLink-conforming XML element that asserts the existence of a link. When an XLink link associates a set of resources, those resources are said *to participate in the link*. Even though XLink links must appear in XML documents, they are able to associate all kinds of resources, not just XML-encoded ones. Using or following an XLink link for any purpose is called *traversal.* Even when the link associates arbitrary numbers of resources, traversal always involves a pair of resources or portions of them. The source from which traversal is begun is the starting resource or portion of it and the destination is the ending resource or a portion of it. In XLink, information about how to traverse a pair of resources, including the direction of traversal is expressed by an *arc*. If two arcs in a link specify the same pair of resources, but they switch places as starting and ending resources, then the link is *bi-directional* (the term used in this context in XLink specification is multidirectional), which is not the same as merely "going back" after traversing a link. *A local resource* in an XML element that participates in an XLink link is specified by value as shown in the following example:

<NAME xlink:type=”resource”>Advanced Multimedia Center</NAME>

A remote resource is specified by URI reference:

<HOMESITE xlink:type=”locator” xlink:href=”http://amc.pori.tut.fi/”/>.

An arc that has a local starting resource and a remote ending resource goes *outbound*, that is, away from the linking element. If an arc's ending resource is local but its starting resource is remote, then the arc goes *inbound*. If neither the starting resource nor the ending resource is local, then the arc is *a third-party arc*. XLink defines *multiple linking* i.e. links with multiple endpoints and/or with multiple starting points. Multiple links connect not only two, but a set of related resources. A link with multiple endpoints is called a multi-ended link. Documents containing collections of inbound and third-party links are called *link databases*, or *linkbases*.

In this work the term *link management* includes link authoring, link storage, link retrieval, link delivery, link maintenance, re-using link structures and link archiving. In *open hypermedia architecture*, links are stored and managed separately from the content (Grønbæk and Trigg 1999).

Time-sensitive links are functions of time as defined in Paper 3 and discussed in the Paper 4. They can be activated by a user when a certain temporal rule or a set of temporal rules are valid. *XLinkTime* is a model for time-sensitive linking structures and it is introduced in the Paper 5. Proof-of-concept implementations of XLinkTime are presented in the Paper 5 and in the Paper 6. *A proof-of-concept implementation* is evidence that demonstrates that a model or an idea is feasible.

In this work, the term *information* is used as data, a single piece of information that has been processed and presented in a form suitable for human interpretation. *A document* is a logical unit of recorded data, which can be presented meaningfully for one or more human beings in at last one socio-organizational context (Päivärinta 2001). For example, a resource on the Web can be an XML document. A useful portion of that resource might be a particular element inside the document. In this thesis examples of resources include project documents and technical manuals. Examples of resource portions are short textual parts of technical manuals, such as step-by-step instructions.

Allen's relations between two time intervals are applied to describe temporal characteristics of documents (Allen 1983).

The term *context* is used in this work to describe circumstances that form the setting for an event. For example, user groups, tasks, situations, environmental aspects, delivery platforms, time and their combinations define different kinds of contexts for using technical manuals. Time is one dimension of context.

The phrase *professional application on the Web* is used in this thesis to refer to operations that support carrying out tasks at work in the World Wide Web environment. Examples of professional applications on the Web are Web-based distributed project management, Web-based technical documentation and Web-based use of technical manuals. The terms *application area* and *application domain* are used as synonyms to application. When designing professional applications to be used on the Web the most relevant use contexts, the amount of resources and the number of associated links between the resources are limited and can be identified and specified in the application design phase. A user's information needs are task-situation oriented and navigation directed.

Distributed project management concerns projects that involve numerous teams situated in various locations geographically. The teams use a common Web-based project and document management system for information interchange and for monitoring the progress of the project. *Technical manuals* describe structural, operational and safety issues of a technical object. A technical object may be a machine, a technical system or a technical product. *A maintenance schedule* of a machine is defined by service intervals such as 8-hour maintenance, 400-hour maintenance and 2400-hour maintenance. Each service interval has its own maintenance instruction procedure.

The *Semantic Web* is an idea of World Wide Web inventor Tim Berners-Lee, director of the W3C (Berners-Lee 1999; Berners-Lee et al. 2001; W3C 2004d). The word "semantic" in the context of the Semantic Web means "machine-processable". Berners-Lee explicitly rules out the sense of natural language semantics. The aim of the Semantic Web is to have data on the Web defined and linked in such a way that it can be used by machines, not just for display purposes, but also for integration and reuse of information.

Metadata is data about data. In the Web environment metadata is machine understandable information for the Web. *Resource Description Framework* (RDF) (W3C 2004c; W3C 2004e) provides a standard way for using XML to represent metadata in the form of statements about properties and relationships of items on the Web. Items, which are known as resources, can be almost anything, with a Web address. Metadata can be associated with a Web page, a graphic, an audio file, or with a movie clip.

Web Service (W3C 2003a; W3C 2004g; W3C 2004h) is a software system designed to support interoperable machine-to-machine interaction over a network.

1.5 List of Publications and the Author's Contributions

The main contribution of the thesis consists of developing a time-sensitive linking structure for professional Web-based applications. The two application domains, distributed project management and technical documentation together with XLink's stages of development during 1997 – 2001 have influenced modelling, designing and implementation of time-sensitive linking structures. The milestones of XLink are described in "Online Resource for Markup Language Technologies" (OASIS 2005).

When W3C published the XLink Working Draft in March 1998, it encouraged possibilities for creating more versatile and context dependent linking structures in the Web. However, the XLink's W3C status remained as working draft for three years and there were no editors or browsers available that supported XLink specification. In July 2000, XLink was promoted to the status of receiving the W3C Candidate Recommendation. In June, 27th 2001, W3C published XML Linking Language (XLink) Version 1.0 as the W3C Recommendation. Nowadays, XLink provides an interesting test bed for constructing and evaluating sophisticated linking structures in the Web and has been a source of inspiration for many researchers in the hypermedia research community.

The dissertation includes six papers, which have been published in three edited books and at three conferences.

Paper 1. Heimbürger, A. 1999. A structured link document as a new means for composing and publishing technical customer documentation in extranets and intranets. In: Smith, J. W. T, Ardö, A. and Linde, P. (Eds.) Proceedings of the 3rd ICCC/IFIP Conference on Electronic Publishing, 10 - 12 May 1999, Ronneby, Sweden. Washington, DC: ICCC Press. Pp. 90 - 102.

Paper 2. Heimbürger, A. 1999. Link management in technical customer documentation. In: Käkölä, T. (Ed.) Proceedings of the 22nd Information Systems Research Seminar in Scandinavia (IRIS 22), Enterprise Architectures for Virtual Organisations, 7 - 10 August, 1999, Keuruu, Finland. University of Jyväskylä, Department of Computer Science and Information Systems, Technical Reports TR-21, Vol. 2. Pp. 13 - 26.

Paper 3. Heimbürger, A. 2003. Modelling time-sensitive linking mechanisms. In: Jaakkola, H., Kangassalo, H, Kawaguchi, E. and Thalheim, B. (Eds.) Frontiers in Artificial Intelligence and Applications, Vol. 94, Information Modelling and Knowledge Bases XIV. Amsterdam: IOS Press. Pp. 26 - 42.

Paper 4. Heimbürger, A. 2005. Developing time-sensitive hypertext linking and navigation support. In: Nicholson, P., Thompson, J. B., Ruohonen, M. and Multisilta, J. (Eds.) E-Training Practices for Professional Organizations. Boston: Kluwer Academic Publishers. Pp. 291 – 298.

Paper 5. Heimbürger, A. 2003. Contextual multi-ended linking: Developing a method for content adaptation. In: Auer, M. and Auer, U. (Eds.) Proceedings of the International Workshop on Interactive Computer Aided Learning (ICL 2003), Learning Objects and Reusability of Content, 24 – 26 September, 2003, Villach, Austria. Pp. 1 - 25.

Paper 6. Heimbürger, A. 2005. Time-sensitive relationship management in technical manuals. Case: Maintenance schedules. In: Kiyoki, Y., Wangler, B., Jaakkola, H. and Kangassalo, H. (Eds.) Frontiers in Artificial Intelligence and Applications, Vol. 121, Information Modelling and Knowledge Bases XVI. Amsterdam: IOS Press. Pp. 152 – 169.

In Paper 1 the author proposed a model for composing and publishing technical customer documentation in extranets and intranets. The model is based on a link document defined by XLink. The link document has two roles in customer documentation. Firstly, it defines the components and the structure for specific customer documentation. Secondly, a link document operates as a user interface for customer documentation. Links as structuring elements in hypermedia applications are discussed in Section 2, Section 3 and Section 4.

In Paper 2 the author described the characteristics of an inter-organisational industrial project. The main actors, their tasks in managing technical customer documentation, and the components of technical customer documentation were analyzed. The author presented a scenario for managing technical customer documentation in extranets from a link management point of view. Development potentials included systematic working procedures, document reuse, content production and composing, testing, archiving, link authoring and transformation from extranet to intranet. Technical documentation is one application area for hypermedia. Related issues are discussed in Section 5.

In Paper 3 the author introduced a concept of time-sensitive linking structure, which is based on Allen's relations between two time intervals and on extended links defined by XLink specification. Extended links include links with multiple destinations, traversal rules between resources and linkbases. Technical customer documentation is discussed as one application domain. The definition of a time-sensitive linking structure is introduced in Section 4.

In Paper 4 the author discussed two approaches, bottom-up and top-down, to create a dynamic workflow chart with time-sensitive linking support for training time-based project management in distributed Web-based environments. The approaches are discussed in Section 4.

In Paper 5 the author introduced an XLinkTime model for time-sensitive linking structures. The XML based syntax of XLink's multi-ended link structure is extended with temporal attributes. Temporal attributes are embedded in an XLink linking element. Although the idea of time-sensitive linking can be regarded as general, the explicit syntax of time-sensitive XLink linking structures are always application dependent. The implementation requires a time analysis of the application concerned. For example, the time-sensitive use requirements of technical manuals are different from the use requirements of project documents. Allen's relations provide a starting point for describing and expressing temporal rules. Two proof-of-concept implementations were described. The latter demonstration is described in Section 5.

In Paper 6 the author constructed a time-sensitive linking application for the instruction manual of a rock grinder. Time conditions were defined by the maintenance schedule of the rock grinder, which includes seven service intervals. Each service interval consists of certain maintenance procedures and tasks. The author demonstrated XLinkTime by embedding the temporal rules in XLink linking elements of an XML document and by separating them from the XML documents and storing them into a separate linkbase file. The proof-of-concept implementation is described in Section 5.

1.6 The Structure of the Thesis

The basic approach of this thesis is more practical and application domains oriented than theoretical and formalism oriented. Hypertext and hypermedia link management research provide the broader background for the thesis (Figure 4). In the past, stand-alone hypertext environments had rich linking possibilities such as multiple linking and separating links from the contents, but they were lost in the World-Wide Web. Many efforts have been made by the W3C and several activities are still on going for the realization of more versatile linking functions on the future Web.

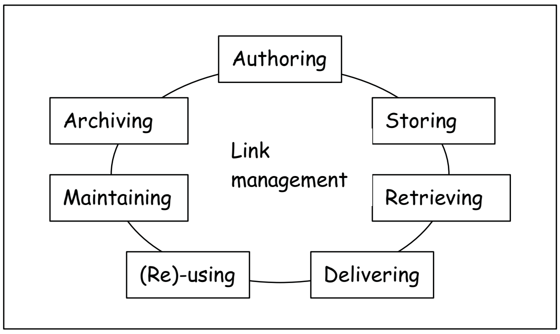


Figure 4. Link management issues

The structure of the thesis with references to Sections and Papers is illustrated in Figure 5. Open hypermedia research, certain application domains, ISO's and W3C's work has all affected the development of XML Linking Language. Based on time models, concepts of multiple linking and link traversals it is possible to develop a model for time-sensitive linking structures, namely XLinkTime. Extending XLink with temporal attributes provides one possibility of realizing XLinkTime. Proof-of-concept implementations were carried out in two ways: by embedding the temporal rules into XLink linking elements and by storing the rules into an XLink linkbase. The time ontological approach was also briefly discussed.

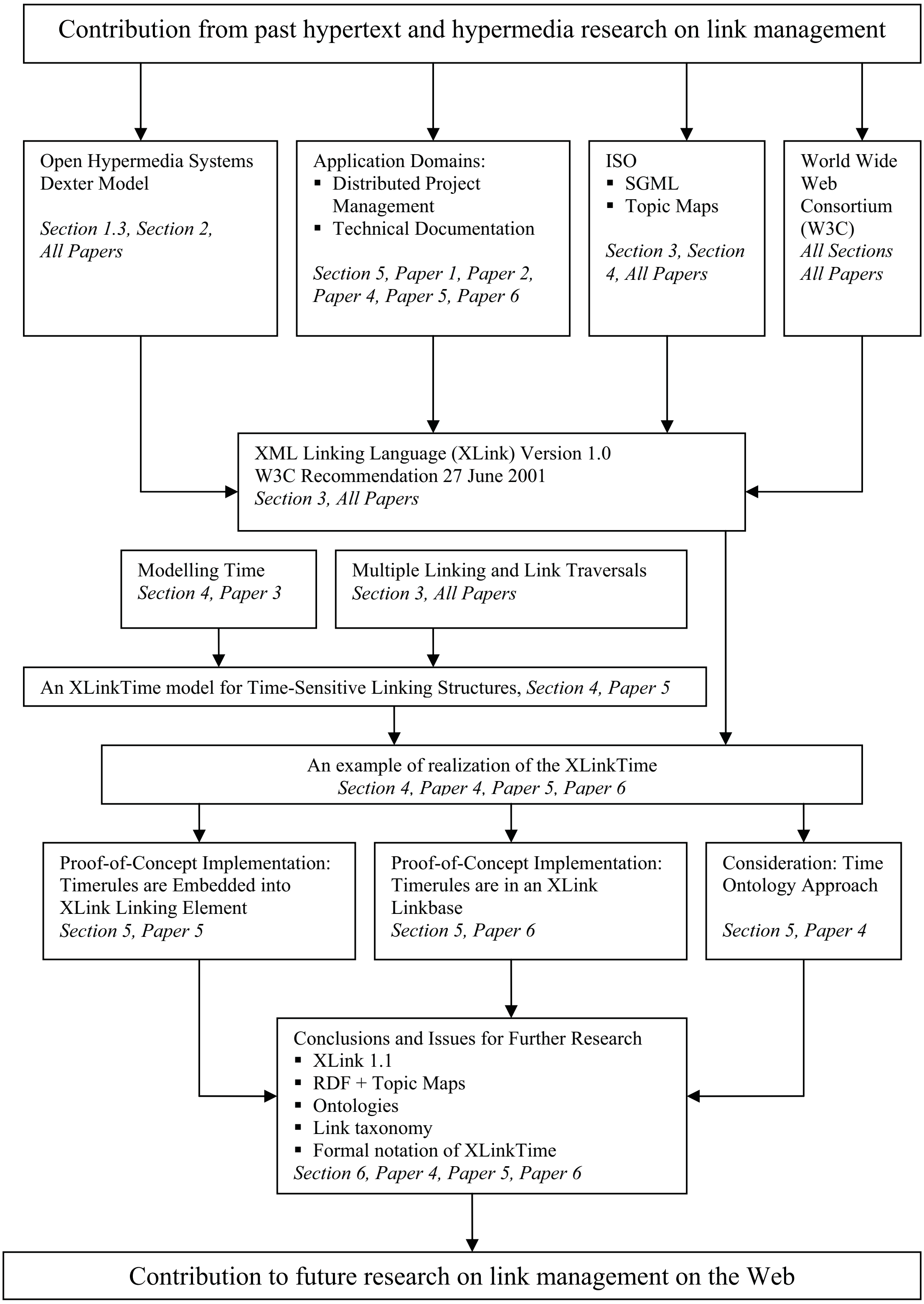


Figure 5. The structure of the thesis

Related work is reviewed in Section 2. Section 3 concentrates on the XML Linking Language. Temporal relations are discussed and a model for time-sensitive linking structure, XLinkTime, is introduced in Section 4. An example of the realization of an XLinkTime model is also given in Section 4. Section 5 describes proof-of-concept implementations of XLinkTime in two application domains: augmenting time-based project management training by means of a dynamic workflow chart and time-sensitive composing of maintenance instructions in the field of technical document management. An evaluation of XLinkTime is also presented in Section 5. Finally, Section 6 is reserved for results, discussion and issues for further research.

Section 2

Related Work

Hypertext, hypermedia and links are the core concepts of the way we create, read, write, manage and access information in the electronic era. Hypertext links appear everywhere on the World Wide Web. Links can enable straightforward, optional or context-based access between resources and/or portions of resources. Hypertext links have changed our ways of working, reading and writing from linear and hierarchal oriented to more associative, relationship emphasized work. Links can be studied from different viewpoints as shown in Figure 6. From the viewpoint of this thesis the development of hypermedia architectures, links in open hypermedia systems, temporal linking and links in context are the essential issues as far as related work is concerned.

The purpose of this section is to complement related work presented in each research Paper and to indicate that there exists no straightforward approach or solution based on the W3C Recommendations to construct time-sensitive linking structures for professional Web applications which will fulfil the premises presented in the previous section.

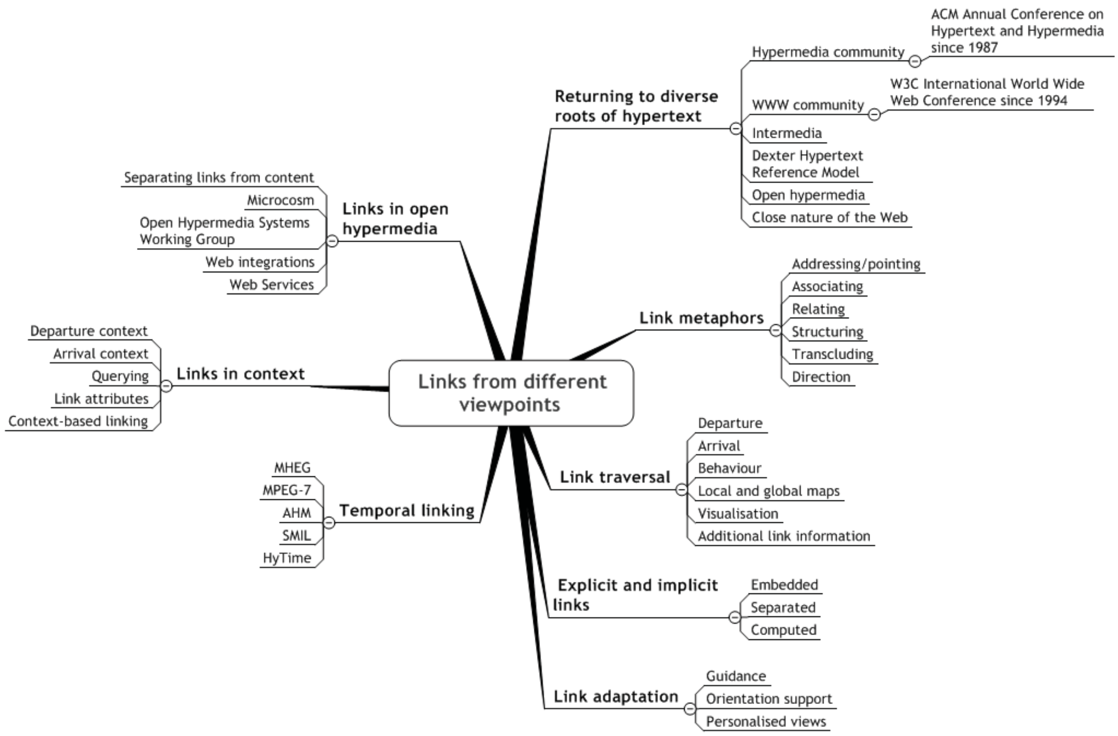


Figure 6. Links from different viewpoints

2.1 Hypermedia and World Wide Web Research Communities

In hypertext and hypermedia literature there are two main research and development communities: the hypermedia community and the World Wide Web community. One of the most essential forums for the hypermedia community to publish their research is ACM's (Association for Computing Machinery) Annual Conference on Hypertext and Hypermedia, beginning in the year 1987. Scientific and technological research, development and communication among members of the Web community concentrate on the W3C forum and its International World Wide Web Conference series, the 17th to come in 2006, as well as also focusing on journals such as Computer Networks and ISDN Systems. These two communities were rather apart, particularly in literature published during the 1990's. During this millennium merging has begun and this is reflected, for example, in co-authored research papers by both communities, and as converging use of hypermedia models, hypermedia formats, and W3C recommendations. Nowadays hypermedia literature is mainly scattered throughout both ICT (Information and Communication Technology) literature and educational literature. However, the above mentioned publishing forums are still the primary sources for the latest research results, both basic and applied, as far as hypermedia and link management are concerned.

In order to understand and follow the ongoing scientific debates and discussions among today's hypermedia and Web communities, it is important to be familiar with hypermedia research before the Web. It is absolutely necessary when participating in scientific research work.

As Grønbæk and Trigg (1999) highlight, hypermedia has grown primarily from English literature and computer science researchers. The World Wide Web emerged from a group of globally dispersed physicists trying to share information. While both fields built upon interrelationships, hypermedia researchers got over 20 years of a head start on designing high-level linking features. On one hand, developers of Web environments can benefit greatly from hypermedia functionality. The hypertext functionality approach encourages system developers to analyse an application's relationships, and how users should access and navigate along these relationships (Bieber et al. 1997). On the other hand, hypermedia researchers are learning tremendously from the Web community about the intensity of demonstrating new technologies, distribution, scale and W3C's process of creating world wide recommendations. The World Wide Web with its structuring, navigation and annotation features is a challenging testing environment for hypermedia functionalities (Bieber et al. 1997).

A selection of forays through the field of hypertext study is reviewed in the ACM Computing Surveys' Electronic Symposium on Hypertext and Hypermedia (Ashman and Simpson 1999). Halasz (1988 and 1991) and Akscyn et al. (1988) discussed seven dimensions that characterize hypertext systems and thirteen issues that defined their vision for those systems. These dimensions and issues are reconsidered in the special issue of ACM Journal of Computer Documentation, Hypermedia Systems in the New Millennium (Waite 2001). One of the main themes in the journal is the role of navigators versus architects. Navigators focus on node contents, whereas architects focus on hypertext structures. The current Web is more for navigators.

Grønbæk and Trigg (1999) pointed out that hypermedia and Web communities represent two rather different philosophies in their understanding of hypertext. In the hypermedia community the Dexter Hypertext Reference Model and its derivatives result from a very deliberate top-down approach, while the World Wide Web community's hypertext represents a rather straightforward bottom-up proposal. The biggest gap between the hypertext philosophies of the two communities originates from the basic principle of the Dexter model: the link and other hypermedia structures should be separated from the data.

In the landmark paper, "Fourth generation hypermedia: some missing links for the World Wide Web" written by Bieber et al. (1997), essential linking issues were discussed. The authors presented a set of high-level hypermedia linking features including typed links, link attributes, transclusions, warm and hot links, private and public links, computed personalized links, external linkbases and link update mechanisms. In the paper the authors discussed implementing requirements for these on the World Wide Web. The biggest problem and obstacle is the close nature of the Web. According to Vitali and Bieber (1999), storing resources and links separately, external linkbases, node and link attributes are the most critical for integrating hypermedia support into the Web environment, and developing the Web towards a more open world-wide hypermedia system.

In this thesis, the concepts of multiple linking, link traversals and linkbases - the linking features which already have existed before the Web - are used to model time-sensitive linking structures. Multiple linking particularly was selected as a starting point because its structure supports the conditional and constraining nature of time-sensitive linking. An example of realizing time-sensitive linking structures will be constructed within the limits of the W3C Recommendations. The time before the existence of the Web and the era of the Web are connected in this thesis in this way.

2.2 Hypermedia Architectures

Hypermedia architectures have developed over the past several decades from closed, monolithic systems towards more open hypermedia systems. The development of hypermedia systems architectures has occurred in four stages (Nürnberg et al. 1997; Nürnberg et al. 1998):

- 1st stage: monolithic systems
- 2nd stage: open link service systems
- 3rd stage: open hyperbase systems
- 4th stage: open hypermedia systems

In each stage of development, some architectural component or function was separated from the hypermedia core. The stages of hypermedia architectures are presented in Figure 7.

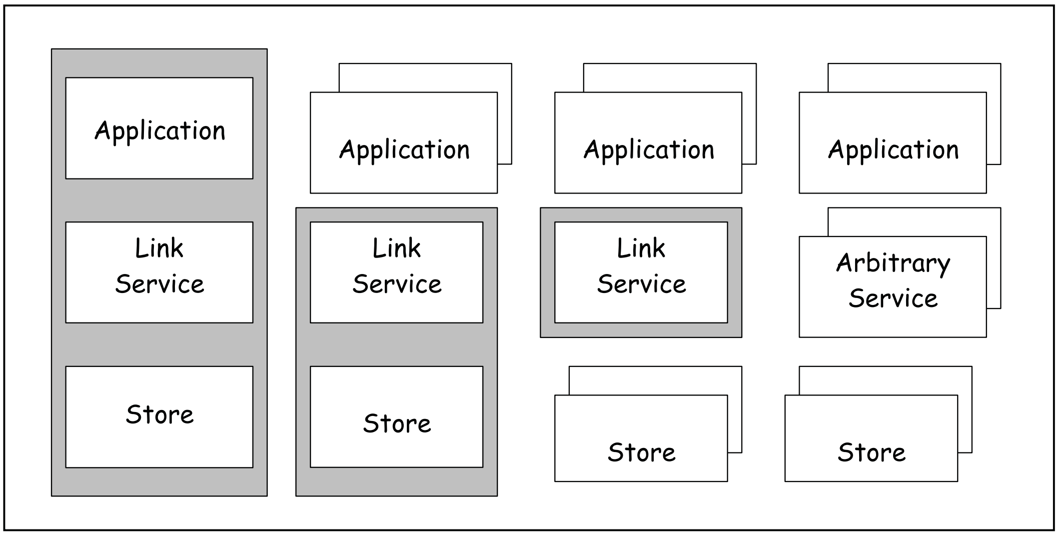


Figure 7. The four stages in the development of hypermedia architectures demonstrate various component separations

After the pioneering work of Bush and Nelson for world-wide web-like systems and Engelbart's work for the development of computer supported co-operative work, there have been lots of research and development activities among the hypermedia community (Bush 1945; Nelson 1987; Nelson 1995; Engelbart 1963; Engelbart 1984). Several monolithic hypermedia systems were created. A good overview and a comparative study of those earlier systems has been reported by Conklin (1987) and Grønbæk and Trigg (1999).

Intermedia was born in the beginning of 1980s by Andries Van Dam and his team at Brown University (USA). Intermedia hypertext system is nowadays one of the classic hypertext systems (Meyrowitz 1989; Van Dam 1988; Yankelovich et al. 1985). The remarkable idea of Van Dam was the "web", a set of links that belongs together. This idea made it possible for different users to create sets of links of their own on the same document. In present hypertext terminology these sets of links are called linkbases.

The development of monolithic systems went on with personal computers. In the mid 1980's Peter Brown invented Guide, the first commercial hypertext authoring systems for personal computers. In 1986 Xerox Palo Alto Research Center released NoteCards. NoteCards supported the creation of graphical overviews of the structure of the hyperdocument. In 1987 HyperCard for Apple MacIntosh was released. The metaphor of HyperCard is based on cards which form stacks. Cards are linked to other cards with link anchors called buttons. Another metaphor for the hypertext system was introduced in Hyperties developed at University of Maryland (Schneiderman 1987). Hyperties used the metaphor of an electronic book or encyclopaedia.

These landmark systems were integrated, in the late 1980's, with the Dexter Hypertext Reference Model (Halasz and Schwartz 1994). The Dexter model combines the best ideas of the systems of the early days with a few conceptual breakthroughs of its own. The second stage in the development of hypermedia architectures began. A number of so-called link services emerged in the late 1980's and early 1990's. A link service is an independent server and it allows the client to connect to a link server to request a set of links to apply to the data in a document (Grønbæk and Trigg 1999). Examples include Microcosm (Fountain et al. 1990, Hall and Davis 1994; Hall et al. 1992; Hall et al. 1996), Devise Hypermedia (Grønbæk et al. 1997), and Chimera (Anderson et al. 1994). In the third developmental stage of hypermedia architectures, the backend of the hypermedia system was separated. Hyperbases provided abstract data models and general database functionality (Grønbæk and Trigg 1999). A hyperbase can support versioning, collaboration and distribution. Examples of an abstract data model and hyperbase management systems were Hypertext Abstract Machine (HAM) (Campbell and Goodman 1988), Hyperform (Wiil and Leggett 1992; Wiil and Leggett 1997a) along with Hyperdisco (Wiil and Leggett 1997b).

The fourth abstraction stage of hypermedia architectures separated link traversal computing and the link server from each other. In the hypermedia community the time of open hypermedia systems had started. However, the World Wide Web (WWW) appeared on the scene in the early 1990's. The Web represents a different line of development in which hypermedia structures are embedded in the data. The simplicity and distributed usage of the Web have brought hypermedia to end users around the world. During the 1990's there were several attempts to merge the ideas of link services and hyperbases with the Web. Chimera, HyperWave, originally called Hyper-G, and Microcosm can be regarded as examples of second generation Web systems, where support for non-embedded links inside and between Web pages was developed (Anderson et al. 1994; Andrews et al. 1995; Carr et al. 1995; Maurer 1996).

None of these systems relies on W3C Recommendations and thus does not fulfil one of the most important premises of this thesis. If we compare the Web-based client-server architecture and the four stages in the development of hypermedia architectures the Web-based solutions presented in this thesis represent the first and the second stages. Solutions that represent the third and the fourth stages will be discussed.

2.3 Links in Open Hypermedia Systems

Hypertext research indicates that a solution to many link management issues is to separate the links from the content (Brown and Brown 1995; Davis 1995). Open hypermedia is an area that has been researched by the hypermedia community for several years and for which a number of systems have been implemented. In open hypermedia systems (OHS), links are managed and stored in special databases called link databases or linkbases. The idea of abstracting links from resources allows for a great deal of flexibility in link maintenance and re-use. Common usage of the Web involves embedding links within resources in the HTML format. In this sense, the Web can be considered as being a closed hypermedia system.

Considerable research on OHS has been conducted in order to provide services for structuring and accessing information. The research on OHS has its roots in two fairly independent threads of research: link server systems (LSS) and hyperbase management systems (HBMS) (Wiil and Leggett 1996). The focus of LSS approaches was to develop middleware components that can provide hypermedia linking functionalities to and from information managed by tools without altering the information itself. In addition to hypermedia linking functionality, HBMS also provides support for content storage (Campbell and Goodman 1988; Wiil and Leggett 1997b).

The idea of a link service has existed since the days of Intermedia (Yankelovich et al. 1988). From a user's point of view, the need for distributed link services has grown with the development of the World Wide Web. Since 1995 the open hypermedia community has been developing various systems for augmenting the Web with externally stored hypermedia structures. Over the past decade, increasingly open, distributed and modular hypermedia systems have been designed by the hypermedia community. Many efforts for open hypermedia and Web integration have been reported in literature (Anderson 1997). The most essential projects are reviewed below.

The Microcosm group did work in this area with several publications. One of the most notable is known as the Distributed Link Service (DLS) (Carr et al. 1995; Carr et al. 1998). Microcosm distinguishes the data that describes links from the link service which supports querying, creating and maintaining the link data (Carr et al. 1998). The term linkbase describes a set of links in Microcosm terminology. A link service may operate with multiple linkbases. Microcosm supports a query-oriented mode of interaction, i.e. a user selects a text string and requests all available links, as a result he/she gets a list of links (De Roure and Hall 1997). De Roure and Hall (1997) classified linkbases as being internal and as external. An internal linkbase overlays a particular structure onto the resource and defines a user's navigation through the assets contained within the resource. An external linkbase takes readers to related resources in other resources, or brings readers from other resources into the one at hand.

Systems like Hyper-G (later HyperWave) (Andrews et al. 1995) and Atlas Link Service (Pitkow and Jones 1996) incorporate concepts that avoid broken links in distributed environments by immediately removing them when a broken link is detected by the server. Hypermedia is often viewed as being either a paradigm for human-computer interaction or for information organization. Nürnberg et al. (1996) discussed a third view, hypermedia as a computing paradigm. They developed a hypermedia operating system implementation (HOSS). Ted Nelson's Xanadu and ZigZag also are examples of hypermedia operating systems (Nelson 1987; Nelson 1999).

Grönbæk et al. (1997) described a distributed link service mechanism based on the Dexter model and they extended it to the Web environment. In this mechanism the links are maintained by a separate server, but combined with the text document by a Java applet embedded in the user's browser. De Roure et al. (2000) introduced a large scale, dynamic and openly distributed link service. Other distributed link services on the Web environment are the Aquarelle project (Rizk and Sutcliffe 1997), Arakne (Bouvin 2000), Chimera (Anderson 1997; Anderson 2001), HyperScout (Weinreich et al. 2001), OHRA (Goose et al. 1997), WebDAV, (Whitehead and Wiggins 1998) and Webvise (Grønbæk et al. 1999).

Open Hypermedia Systems Working Group (OHSWG) was established in order to address interoperability problems between OHS's (Davis et al. 1999; Grönbæk and Wiil 1997; Nürnberg and Leggett 1997). The OHSWG constructed an open hypermedia protocol (OHP) which is based on the Dexter model (Reich et al. 1999). OHP defines a standard way for any OHP-aware client to communicate with any OHP-aware link server.

New trends in OHS research on the Web environment are developing from structure servers. Auld Linky is an example of a structure server, which is capable of handling structures such as trails and tours in addition to links (Millard et al. 2003). Walden's Paths supports trails along Web pages (Furuta et al. 1997; Dave et al. 2003), and Ariadne provides guided tours (Jühne et al. 1998).

Nürnberg and Ashman (1999) raised the fundamental question of interaction between the World Wide Web and open hypermedia research communities. In their thesis - antithesis argument, the Web is an open hypermedia system – the Web is not an open hypermedia system. They ended up with two conclusions: (a) the Web can be made to emulate an OHS by Web programming and (b) the Web cannot be an open hypermedia system, because it is not even a hypermedia system and thus its openness is an issue that need not be addressed.

Research on and implementations of open hypermedia systems have produced several interesting environments. However, these systems basically rely on the data formats developed by the open hypermedia research community, such as Fundamental Open Hypermedia Model (FOHM) (Millard et al. 2000) or Open Hypermedia Interchange Format (OHIF) (Grønbæk et al. 2000). The premise of this thesis is to use W3C Recommendations.

2.4 Temporal Linking

Hypermedia applications can contain static and dynamic links and hence be statically and/or dynamically organised into one or more structures. The source and the destination of a static link are unambiguous, defined by the author of the hypertext. Static links can be embedded in or separated from the content. Dynamic links are computed on the fly, and are never explicitly present in the hypertext. Combinations of static and dynamic links have been used, for example in constructing hypermedia learning environments (Multisilta 1996).

A time-based presentation is composed of a number of different media items, such as text, sound, animation and video sequences. Each item has its own duration. If the presentation proceeds in a linear way, this is called multimedia. When the media items are combined into a presentation in a more associative way it is called hypermedia. In both cases time rules are specified by temporal relationships among the different items. Temporal relations define temporal dependencies amongst media items. These relationships are often in the form of constraints. Hardman and Bulterman (1997) and Hardman et al. (1999) define temporal composite as a collection of constrained media items. They define intra-object synchronization as linking within a linear multimedia presentation and inter-object synchronization as linking among multimedia presentations. The main timing issue in multimedia and hypermedia presentations concerns synchronizing, i.e. presenting media items in a certain order.

Spatio-temporal synchronization issues in composing multimedia and hypermedia presentations have inspired many research groups in the hypermedia and multimedia research community (Blakowski and Steinmetz 1996; Bouvin and Schade 1999; Buchanan and Zellweger 1992; Buchanan and Zellweger 1993; Courtiat et al. 1996; Cunliffe 2000; Soares et al. 2000; Georganas et al. 1996; Hall 2000; Hardman et al. 2000;

Hardman et al. 1993a; Hardman et al. 1993b; Hardman et al. 1999; Nanard and Nanard 1995; Nanard et al. 2000; Nielsen 1995; Pérez-Luque and Little 1996; Shipman and Hsieh 2000; Tong 2001; Vazirgiannis 1999). In the computer science research community, some Petri net -based models have also focused on multimedia data stream synchronisation (Sénac et al. 1995; Stotts and Furuta 1989; Stotts and Furuta 1990; Wang and Wu 1995).

Their work has influenced the development of multimedia and hypermedia standards and specifications related to spatio-temporal synchronization issues in composing multimedia and hypermedia presentations. The most essential characteristics of international standards and specifications are briefly presented next. The purpose is to indicate that none of these standards or specifications is directly applicable or usable in the research problem of this thesis.

MHEG (Multimedia and Hypermedia Information Coding Expert Group)

MHEG was set up by ISO (ISO/IEC 13522-1. 1997; ISO/IEC 13522-8. 2001). The task of the group was to create a standard method for the storage, display and exchange of platform independent multimedia presentations between different machines. Since the development basis of the MHEG standard family concentrates on multimedia presentations and digital television broadcasting, there is no research interest for deeper study from the point of view of this thesis.

MPEG (Moving Picture Experts Group)

MPEG is another standardization group that has been working with interactive multimedia and digital television. The MPEG committee developed the standards known as MPEG-1, MPEG-2, MPEG-4 and MPEG-7 (Nack and Lindsay 1999a; Nack and Lindsay 1999b). MPEG-7, formally “Multimedia Content Description Interface” provides a set of standardized tools to describe audio-visual contents, i.e. quantitative measures of audio-visual features, description of structures and relationships (Martinéz 2002). The descriptions of temporal information in MPEG-7 are based on the ISO 8601 standard (ISO 8601. 2000). MPEG’s Time and MediaTime describe time information in the real world and in media streams, respectively. In MPEG, a time instant, t_1, can be described by a Time Point. An interval, $[t_1, t_2]$, can be described by its starting point, t_1, using the Time Point and a Duration, $t_2 - t_1$. The specification of time can also be done with a predefined interval called Time Unit and by counting the number of intervals. This specification is particularly efficient for periodic or sampled temporal signals. Links are simple and they are represented by an element of MPEG-7’s own Description Definition Language (DDL) (Martinéz et al. 2002) (Figure 8).

```
<MediaLocator>
        <MediaUri>http://www.xy.org/mpeg/logo.gif/</MediaUri>
</MediaLocator>
```

Figure 8. An example of a link in MPEG-7’s own Description Definition Language

MPEG-7 is a metadata standard. Its focus is to describe features of audio-visual contents. It uses simple links and a Description Definition Language of its own. For these reasons MPEG-7 is out of the scope of this thesis

AHM (Amsterdam Hypermedia Model)

AHM extends the Dexter model by adding timing and presentation relationships found in multimedia presentations (Hardman et al. 1994; Hardman et al. 1993b; Hardman 2000). The AHM is based on the CWI Multimedia Interchange Format (CWI = Centrum voor Wiskunde en Informatica) (Hardman 2000). The format allows authors to specify how individual pieces of information relate to each other over a period of time. The temporal compositions can be parallel or sequential (Hardman 2000). AHM has been implemented in the Madeus system which is a multimedia authoring and presentation tool (Hardman 2000). The focus of AHM is on synchronizing multimedia and hypermedia presentations, and it uses a format of its own.

SMIL 2.0 (Synchronized Multimedia Integration Language)

The Amsterdam Hypermedia Model has influenced the developmental work of SMIL 2.0 at the W3C. SMIL 2.0 defines an XML-based language that allows authors to write interactive multimedia presentations for the Web (Jourdan 2001; W3C 1998b; W3C 2005c). Using SMIL 2.0, an author can describe the temporal behaviour of a multimedia presentation, associate hyperlinks with media objects and describe the layout of the presentation on a screen. SMIL 2.0 is defined as a set of markup modules such as media object modules, layout modules, timing and synchronization modules and linking modules. These modules define the semantics and XML syntax for SMIL functionality. SVG (Scalable Vector Graphics) uses SMIL timing to create animation effects (W3C 2005a).

The SMIL 2.0 Timing Module defines elements and attributes to coordinate and synchronize the presentation of media over time (Bulterman 2001; Bulterman 2002). The SMIL Timing Module also provides attributes that can be used to specify an element's timing behaviour. Elements have a beginning, and a simple duration. The beginning can be specified in various ways - for example, an element can begin at a given time, or begin based upon when another element begins, or begin when some event (such as a mouse click) happens. Simple duration defines the basic presentation duration of an element. Elements can be defined to repeat simple duration, a number of times or for a certain amount of time.

SMIL 2.0 defines the anchor element, an *area* that allows breaking up an object into temporal subparts. In the following example links are associated with temporal segments by means of the area element (Figure 9) (W3C 2005c). The duration of a video clip is split into two sub-intervals. A different link is associated with each of these sub-intervals.

```
<smil
xmlns="http://www.w3.org/2001/SMIL20/Language">
  <body>
   <video src="video" title="Interview" >
      <area begin="0s" dur="20s" title="first question"
          href="http://www.example.org/>
      <area begin="20s" dur="50s" title="first answer"
          href="http://www.example.org/>
   </video>
  </body>
</smil>
```

Figure 9. Links are associated with temporal subparts by means of the area element

The SMIL 2.0 Linking Modules define navigation means through the SMIL presentation. SMIL 2.0 links are limited to uni-directional single-headed links, i.e. all links have exactly one source and one destination resource. The SMIL 2.1 W3C Candidate recommendation dated 13 May 2005 leaves the SMIL 2.0 Linking Modules unchanged (W3C 2005c; W3C 2005d).

One of the premises in this thesis is to consider multiple linking structures.

<u>HyTime (Hypermedia/Time-based Structuring Language)</u>

HyTime provides standardized mechanisms for specifying links, within and between resources, and for scheduling multimedia information in time and space (Derose and Durand 1994; ISO/IEC 10744. 1997; Newcomb et al. 1991). HyTime is based on the SGML (Standard Generalized Markup Language) standard.

HyTime defines multi-ended link and it also provides attributes for controlling traversal among hyperlink anchors. An application can specify the traversal that is permitted with respect to each anchor by means of the link traversal rules (*linktrav*) attribute. This attribute specifies whether or not it is an initiation anchor and what traversal is allowed for each anchor.

The scheduling module of HyTime defines the architecture for abstract representation of hypermedia structures by means of sequencing objects along axes measured in temporal and/or spatial units.

HyTime standard is very extensive in providing facilities for representing linking and scheduling of media items in space and time. The disadvantage of HyTime has been its generality and, as a consequence, a lack of supporting tools and implementations. However, HyTime includes the concepts of multiple linking and link traversal rules. HyTime has influenced the development of the W3C XML Linking Language (XLink), which will be introduced in Section 3.

Three other interesting research papers on temporal linking are worth mentioning. XConnector is a project where XLink was extended to provide multimedia synchronization facilities for Web resources with a template approach (Muchaluat-Saade and Soares 2002).

Beales et al. (2001) and Page et al. (2001) investigated supporting link creation for continuous media. The first application of their HyStream system was to deliver seminar resources in the form of video, and a related set of slides. The titles of the slides served as hypertext links associated to the video sequences. The predominant view of temporal media is that of a discrete block within a hypermedia system. It is the synchronised presentation of the media with other hypermedia elements that has received the most attention. As such, the mechanisms "to jump to and from" particular time indexes within media are used. The timeline is superimposed so that hypermedia elements can be synchronised. For example, in HyStream a point in the video can be accessed as a link to the video itself, appended with a time index.

Beales et al. (2001) and Page et al. (2001) noted that the situation becomes more complicated when temporal media is unbounded, such as in a live scenario. It becomes more difficult to deal with the media as a block within a carefully scripted hypermedia presentation, because the block has an infinite length. They discussed how time could be represented. One way would be to introduce time as a separate entity to which other entities are related. Another alternative might be to use time as a context. Viewing an entity from a particular time context could resolve that entity with the correct data will be shown.

Sumiya et al. (2001) investigated temporal linking in the context of hypermedia broadcasting where contents depend on time. Contents have transaction time and validity time. Transaction time is the time at which the content is registered in the database. Validity time is the time for which the contents are accessible. Push-type services were studied. In the push-type service, the system periodically delivers information to each user who has registered in advance to a channel or another source of interest. In the Mille-feuille prototype system for push-type services Sumiya et al. (2001) have used XML and they defined four files. The transaction time and validity time were added as the attributes of the dtd-file. In source.xml and destination.xml files these time attributes are set according to the ISO 8601. Link sources and link destinations are defined in link.xml files which use XLink's multiple link syntax. In this approach time parameters are defined in nodes.

The papers by Beales et al. (2001), Muchaluat-Saade and Soares (2002) and Page et al. (2001) discuss time as a context and associating correct data to a certain time context. However their approach emphasises synchronising hypermedia presentations, i.e. presenting media objects in a certain order. The paper by Sumiya et al. (2001) discusses the possibility of extending their approach by setting time parameters in the link itself in push-type services. This idea is very close to the interests of this thesis. However, in this thesis resources do not depend on time as they do in push-type broadcasting environments, and thus the approach to define time parameters in nodes is not useful.

2.5 Links in Contexts

In hypermedia literature three different, partly overlapping, approaches to considering context can be identified:

- context is defined in the sources and destinations of links
- context-dependent information access according to user profiles
- information grouping according to context.

A consequence of associating a context with sources and destinations of links, is that the same anchor can be used in different links with different contexts. Link contexts help authors organize information more effectively and indicate different views for users.

Source and destination contexts for a link are defined as part of the Amsterdam Hypermedia Model (Hardman et al. 1993a). The context makes explicit which part of a presentation is affected when a link is followed from an anchor in the presentation (Hardman et al. 1993b).

Rich links, i.e. links with attributes, have been utilised in knowledge-intensive domains in order to complement basic navigation (Oinas-Kukkonen 1998; Oinas-Kukkonen 2000). Examples of such domains are software engineering (Oinas-Kukkonen 2000) and collaborative design in distributed authoring and versioning environment (Takahashi 1998).

Information access in context allows users to search for information that is directly or indirectly available in the document that a user is currently reading. A Web page, such as a Google search result with its direct links to documents and "search similar documents" –link command, is an example of this. Staff (2002) presented a HyperContext framework which is based on paths selected by users in hyperspace. As users navigate a HyperContext hyperspace they can create links between resources. Whenever a link is created, the user is asked to describe why the destination resource is relevant to him or her. The basic starting point is that information in the same resource can mean different things to different users. In HyperContext the different descriptions of information are called interpretations. Hirashima et al. (1997) introduced a filtering method for context-dependent browsing. The filter takes advantage of a user's interests from the previous browsing history.

Na and Furuta (2000) developed a context-aware hypertext model, caT, in order to support flexible adaptation in dynamically changing environments. The caT model integrates Trellis, which is a Petri net based model of hypertext (Stotts and Furuta 1989, Stotts and Furuta 1991), and user modelling. In the Trellis model, hypertext is considered as having two layers. A fixed underlying information structure was created by the hypertext author. A flexible structure can be generated dynamically according to the user's requirements. The caT model considers that the relevance of information to a user may vary with time of day or location. Also the presentation of information may differ depending on device and network characteristics (Furuta and Na 2002). Authoring caT and Trellis hypertexts starts from structure and then proceeds to the content. The link graph structure of the hypertext was developed first and then the content was associated with the nodes. Content is created in the Web first and then the links are embedded into the content. According to Furuta and Na (2002), one might argue that it is the node that is the primary abstraction within the Web while it is the link structure that is primary in caT.

Auld Linky, the beta version of which is known as Auld Leaky (Michaelides et al. 2001; Weal et al. 2001), is a structure server that is specially designed to be used as an OHS with contextual support. Contexts in Linky resemble user profiles essential in adaptive hypermedia research (Brusilovsky 1996; Brusilovsky and Maybury 2002;

Brusilovsky et al. 2002; Ketamo 2002) more than they resemble contexts or linkbases in open hypermedia works. Auld Linky is a stand alone process that manages an XML linkbase of association structures expressed according to the FOHM and provides pattern matching services via HTTP (Hypertext Transfer Protocol). An example of a context is given in Figure 10 (Millard et al. 2003).

```
<association id='vdoc001'>
        <relationtype>virtualdoc</relationtype>
        <structure>list</structure>
        <feature>position</feature>
        <binding>
                <featurevalue feature='position'>1</featurevalue>
                <reference>
                        <data><url><! [CDATA [Good-day.xml]]></url>
                                <behaviour><event>ondisplay</event>
                                <behaviourvalue key= 'time'>night</behaviourvalue>
                                </behaviour>
                        </data>
                </reference>
                <context>
                        <contextvalue key= 'time'>day</contextvalue>
                </context>
        </binding>
        …
</association>
```

Figure 10. An example of an association element in Auld Linky context

When users query Linky, they do so within a predefined context. Auld Linky provides a view of all the loaded linkbases that fulfil the context search condition. Auld Linky was demonstrated by generating short stories from a story database and historical notes on the city of Glasgow from tourist information system (Michaelides et al. 2001; Millard et al. 2003).

El-Beltagy et al. (2001) defined a context aware linkbase where links in a given context can be grouped together. The system is based on user profiles and it can automatically generate links that can serve a certain community of users.

Lemahieu (2002) presented a maintainable, end user friendly and structured hypermedia model (MESH). The MESH model is built upon an object-oriented data model and context-based navigation paradigm. Navigation along static links is complemented in MESH by run-time generated linear, guided tours. These are derived dynamically from the context of a user's information requirements. MESH separates direct and indirect links. Direct links are typed and reflect the underlying conceptual data model. Direct links are permanent and context-independent. They can be stored explicitly in the linkbase and they are always valid. Indirect links do not only reflect the data model but also depend on a run-time variable, the current context. They cannot be stored within the linkbase but they are to be created dynamically at run-time.

In the human-centred computing paradigm the user, the task at hand and the context in which the task is accomplished are considered on a global level. Brézillon (2003) separates contextual knowledge and proceduralized context. Contextual knowledge is background knowledge, whereas proceduralized context is immediately useful for the task at hand. An important issue is the transition from contextual knowledge to a proceduralized context – that is the context's dynamic dimension. This contextualization results from focusing on a task at a given step. For example, the operator who controls process monitoring in a subway control room has just a few minutes when an incident occurs to analyze the situation, collect contextual information, identify the context in which the incident occurred and decide what to do. Brézillon (2003) also defines different types of contexts: static, dynamic, discrete, continuous, individual and collaborative contexts, as well as immediate context, such as the environment, and distant context, such as weather forecasts.

In a more general sense, the interpretation of the word context shows considerable variety as Berztiss (2002) pointed out in his analytical article. The definition itself is context-sensitive depending on the definer's field of expertise. The notion of context has been defined in cognitive psychology, linguistics and computer science. According to Berztiss (2002) a general computer science oriented definition for context can be given as follows: context is a pair <**w**, t>, where **w** is a slice of the world at time t (Figure 11).

Context $C = \langle \mathbf{w}, t \rangle = (\langle w_1, t \rangle, \langle w_2, t \rangle, \ldots, \langle w_n, t \rangle)$,
where w_n is a slice of the world at time t. Sometimes **w** is a data set and sometimes it is a set of rules.

Figure 11. A definition of context

In this thesis it is emphasised that when a user accesses Web resources and/or portions of resources, he/she is only interested in some certain aspect of the provided information. He/she knows what kind of information to look for. In the context of the specific problem the user is dealing with grouping Web resources and/or the portion of resources is relevant. In this thesis time is the main dimension of context. Resources are grouped according to a temporal rule or a set of temporal rules related to a certain task and to a certain situation.

2.6 Summary

The parallel development of hypertext research prototypes and the World Wide Web has resulted in repeated attempts to replace the Web or offer world-wide all-purpose services to augment the Web with missing functionalities. The hypertext research community has defined many kinds of links and has proposed many ways to implement them. In particular, a much debated issue has been whether links should be part of resources or whether they should be kept separate. The literature also describes several formal models of hypertext.

The hypermedia systems reported in literature basically rely on the data formats developed by the hypermedia research community. The premise of this thesis is to use W3C Recommendations. These recommendations will be discussed in the next section.

Section 3

XML Linking Language

When W3C published the XLink Working Draft in March 1998, it encouraged possibilities of creating more sophisticated linking structures on the Web. However, the XLink's W3C status remained as a Working Draft for three years and there were no editors or browsers available that supported XLink specification. In July 2000, XLink got the status of W3C Candidate Recommendation. On June, 27th 2001, W3C published the XML Linking Language (XLink) Version 1.0 as the W3C Recommendation (W3C 2001b). The whole development history of XLink is described by OASIS (Online Resources for Markup Language Technologies) (OASIS 2005).

XLink defines extended links with multiple linking, link traversals and linkbase functionalities and as such, introduces the possibility of implementing more versatile link structures, already defined in the earlier days of hypermedia systems such as Intermedia, on the Web. XLink is quite sophisticated and today is applied mainly to customized and professional applications. However, at the moment, XLink is the only formal linking language that is based on the international standard and that has a recommendation status in the W3C (ISO 8879). XLink enriches Web-based linking and provides a test-bed for constructing and evaluating diverse linking structures in Web applications.

The purpose of this section is to discuss HTML linking limitations, describe the main characteristics of XLink and XPointer languages. The section provides necessary background information for the research problems, because XLink provides one possibility for realizing time-sensitive linking structures.

3.1 HTML Linking Limitations

When we consider professional applications the linking mechanism on the Web is weak, although, there are some strengths in HTML linking. HTML has a simple syntax and it is easy to implement. However, there are some linking limitations in HTML.

HTML links are uni-directional. The possibilities for typing links are limited. Link databases are not allowed, which means that link sets cannot be filtered, selected on demand or different sets associated with the same resource. An HTML link can point to a whole resource or to a certain point inside the resource but there is no mechanism to define a specified portion of a resource to be pointed at. There are no multi-ended links in HTML.

3.2 XML Linking Language Family

SGML is ISO's international standard for defining markup languages, and it is designed to promote text interchange (ISO 8879). Markup languages define the markup rules, which add meaning to the structure and content of documents. They are the grammar and the syntax which specify how a language should be "spoken". XML is a subset of SGML, and it is especially designed for the Web (W3C 2004a).

XLink is an XML based recommendation defined by the W3C. XLink specifies how separate documents should be linked to one another, and how structures within XML documents should be addressed. XLink provides more powerful link representation and addressing features than HTML.

The XML Linking Language Working Group of the W3C has produced a family of linking related languages: XLink, XPointer, XPath and XML Base (W3C 1999; W3C 2001a; W3C 2001b; W3C 2003b; W3C 2003c; W3C 2003d). The whole set is called the XML Linking Language family (Figure 12).

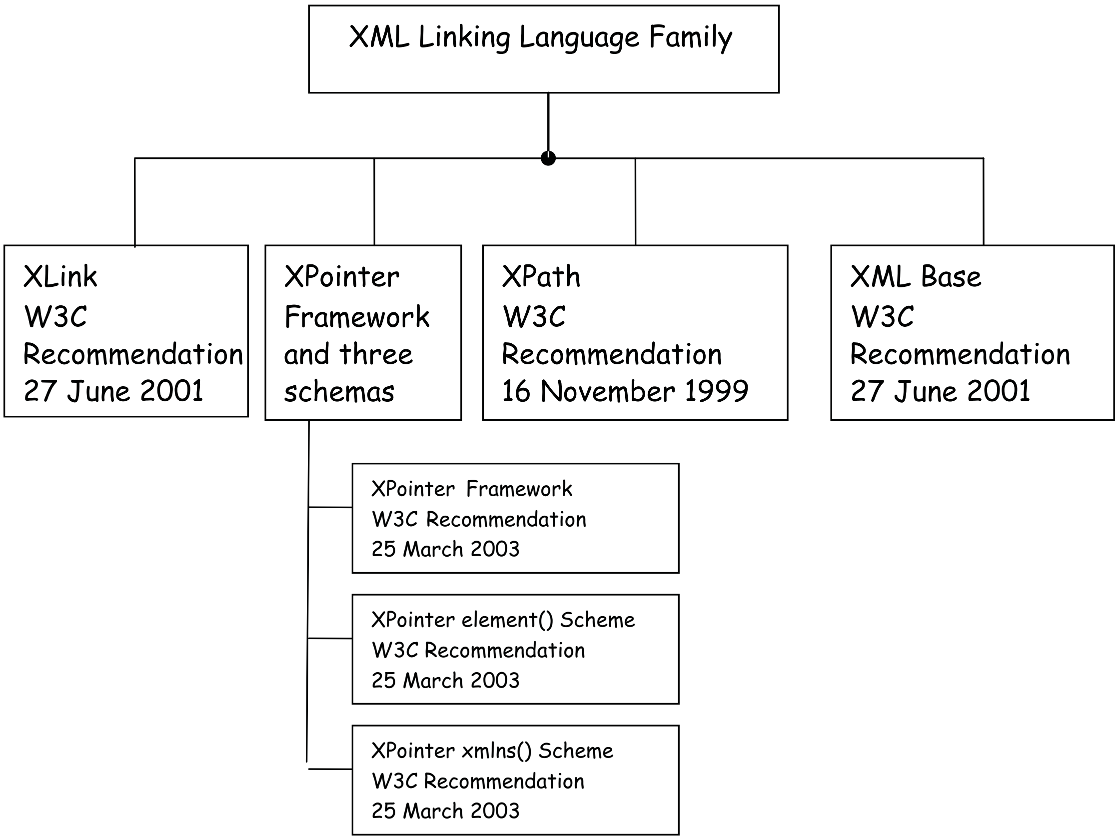


Figure 12. Members of the XML Linking Language Family and their W3C statuses

XLink allows elements to be inserted into XML documents in order to create and describe links between resources. XPointer, based on XPath, identifies and locates parts of XML resources. XML Base defines a mechanism for providing base URI services to XLink, especially for the purpose of resolving relative URI addresses in links to external resources.

3.3 XLink

Many other earlier linking systems have affected the design of XLink, especially Augment, Dexter, FRESS, Intermedia, OHS, MicroCosm, Hyper-G and HyTime. In addition to these projects and models, the Text Encoding and Interchange (TEI) initiative group has also been active (DeRose and Durand 1995). The effort of XLink has been to integrate the strengths of earlier hypertext research, in order to provide more versatile but yet simple linking features. XLink gets many of its core concepts from HyTime, such as multidirectional links, multi-ended links and traversal of links. In the following subsections link structures defined by XLink are introduced, and some examples of XLink implementations are provided.

3.3.1 Attributes of Links

The most important goals of the XML Linking Working Group have been to provide: (a) attributes for links, (b) diverse link behaviour, (c) bi-directional links, (d) multi-ended links and (e) a possibility for linkbases. XLink provides a means for defining syntax for semantic, behavioural and traversal attributes for a link (Figure 13).

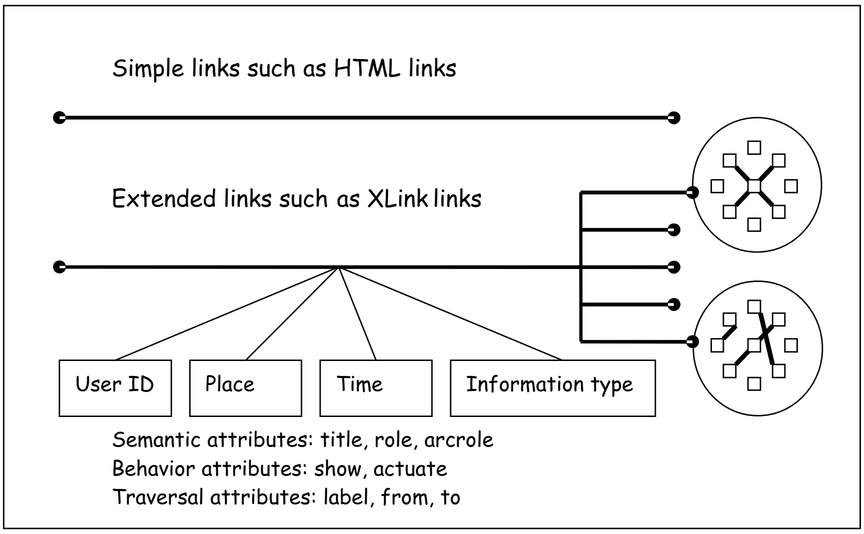


Figure 13. XLink link has attributes to define its functionality

In XLink a linking element is an element that tells of the existence and describes the characteristics of a link. A linking element can have any name, but the way to inform the XLink conforming browser that this element should be treated as a link is done by using the *type* attribute that identifies the XLink element. The *type* attribute indicates the XLink element type (simple, extended, locator, arc, resource or title).

XLink supports simple, unidirectional, unary and untyped links as well as more sophisticated, extended links. Extended links can be bidirectional and *n*-ary. Arcs are used to describe traversal rules in an n-ary link. Participating resources may be in local servers or in remote servers. XLink also defines a linkbase for storing a set of extended links to separate XML files. Extended links can also have attributes. XLink consists of three attribute categories: semantic, behaviour and traversal attributes. The semantic attributes, *role, arcrole* and *title*, describe the meaning of resources within the context of a relationship. The behaviour attributes, *show* and *actuate*, describe a link's behavioural intentions. The *show* attribute can have the values *new*, *replace*, *embed*, *other* or *none*. The values of the *actuate* attribute can be *onLoad*, *onRequest*, *other* or *none*. The third attribute category is traversal attributes, *label*, *from* and *to*. The *label* attribute of the locator or the resource element describes from which link traversal is initiated and to which traversal goes. *From* and *to* attributes define the starting and ending resources of a link's traversal arc. Table 3 summarises the element types (columns) and allowed attributes (rows) of XLink, indicating also if the value is required or optional. No note in a row means that the attribute is not allowed with the type-attribute value under consideration.

Table 3. XLink's element types and allowed attributes

XLink element type Allowed attributes	simple	extended	resource	locator	arc	title
type	required	required	required	required	required	required
href	optional			required		
role	optional	optional	optional	optional		
arcrole	optional				optional	
title	optional	optional	optional	optional	optional	
show new, replace, embed	optional				optional	
actuate onRequest, onLoad	optional				optional	
label			optional	optional		
from					optional	
to					optional	

3.3.2 Multiple Linking

Links with multiple endpoints connect not only two, but a set of related resources. For example, if a user initiates the traversal of a link with multiple endpoints, he/she can be requested to choose between the options available on a menu or in pop-up windows. Multiple links can contain multiple links as well, which makes nested tables of contents or multiple linking networks possible. Examples of two XLinks are shown in Figure 14 (Christenssen et al. 2003).

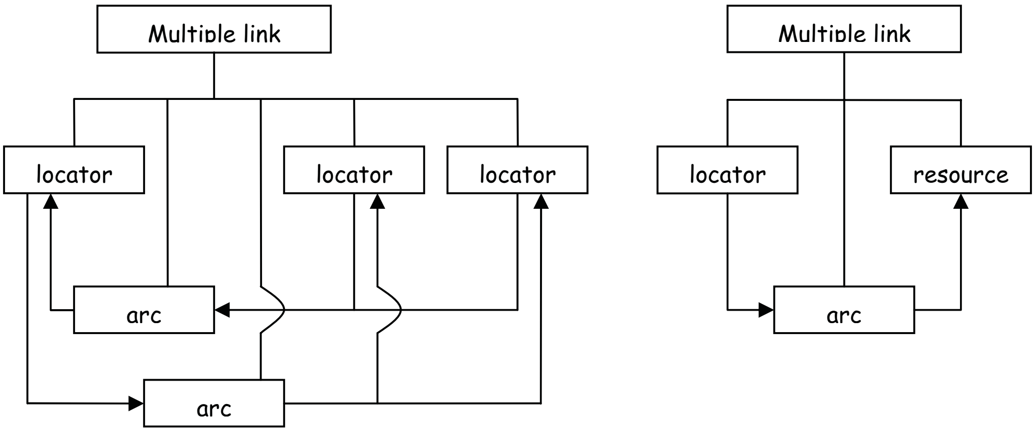


Figure 14. Examples of two multiple links. The link on the left is multi-ended and partly bi-directional. The link on the right is uni-directional, linking a remote resource to a local resource

Figure 15 shows a multiple link that associates five resources and provides traversal rules between them. The resources could be, for example, components of technical customer documentation. The first resource is a description of installation procedures, the second resource represents operation, the third resource contains visuals, the fourth resource concerns maintenance and the last resource represents identified troubleshooting situations.

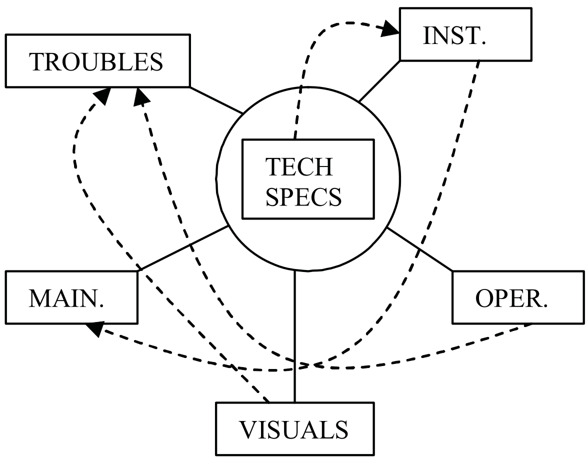


Figure 15. An example of an extended link that associates five resources

Without a multiple link, the resources might be entirely unrelated. They might be in five separate documents, which is not desirable for example in troubleshooting situations where pieces of information from the whole documentation are needed and situation-specific assemblies should be composed. In Figure 15 the non-directional solid lines indicate that the link connects the five resources. The dotted arrows indicate examples of traversal rules that the link structure provides. Traversal rules are necessary because, without them the resources are associated in no particular order with any implication as to whether and how individual resources are accessed.

A multiple link indicates rules for traversing among its resources by means of an *arc* element. The *arc*-type element may have the traversal attributes, *from* and *to*, the behaviour attributes, *show* and *actuate*, and the semantic attributes *arcrole* and *title*. The value of the arcrole attribute must be a URI reference, and the title attribute is used to describe the meaning of a link or resource in a human-readable fashion (W3C 2001b). The traversal attributes define the desired traversal between resources that are connected by the same link. A *from* attribute defines a starting resource, and a *to* attribute an ending resource. The behaviour attributes specify how ending resources will be shown to the user. In Figure 16 is an example of how the syntax of a multi-ended XLink linking element might look.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<links xmlns:xlink=http://www.w3.org/1999/xlink xlink:type="extended" >
<techdocload>
        <tcd>Technical Customer Documentation</tcd>
                <installation
                xlink:type="locator"
                xlink:href="Technical_object/installation.xml"
                xlink:label="installation "
                xlink:role="http://www.example.com/linkprops/installation "
                xlink:title="Installation"/>
                <operation
                xlink:type="locator"
                xlink:href="Technical_object/operation.xml"
                xlink:label="operation"
                xlink:role="http://www.example.com/linkprops/operation"
                xlink:title="Operation"/>
                <training
                xlink:type="locator"
                xlink:href="Technical_object/training.xml"
                xlink:label="training"
                xlink:role="http://www.example.com/linkprops/training"
                xlink:title="Training"/>
                <go
                xlink:type="arc"
                xlink:from="installation"
                xlink:arcrole="http://www.example.com/linkprops/operation"
                xlink:to="operation"
                xlink:show="replace"
                xlink:actuate="onRequest"
                xlink:title="Maintenace:_Code_434" />
                <go
                xlink:type="arc"
                xlink:from="operation"
                xlink:arcrole="http://www.example.com/linkprops/training"
                xlink:to="training"
                xlink:show="replace"
                xlink:actuate="onRequest"
                xlink:title="Trouble_shooting_situation:_Code_434" />
</techdocload>
```

Figure 16. An example of the syntax of a multi-ended link

3.3.3 Linkbases

The idea of external databases is very attractive in Web-oriented environments where the rapid expansion of available data means that hard-coded links in Web pages are very hard to keep up-to-date (Ciancarini et al. 2002). External linkbases are much easier to maintain and manage. Information has to be updated in one place rather than in multiple files.

The more complex and versatile linking structures are, the more an external linkbase solution is recommended, especially in professional applications where broken links are not allowed. Be the links embedded or separated, be the Web present, semantic or mobile, there will always be links that are to be authored according to the syntax of some linking language, delivered via some protocol, used by means of browsers and maintained manually, semi-automatically or by means of a solution to be seen in the future.

XLink provides a way to store the lists of links outside the resources that are connected to each other. These are called linkbases. By means of linkbases, it is possible to maintain several link sets which can be associated with the same set of resources. Linkbase files are XML files separated from the browsed resources and containing only extended XLinks. XLink linkbase files can contain any number of extended links.

The *xlink:arcrole=http://www.w3.org/1999/xlink/properties/linkbase* attribute makes the browser treat the link like a special link, i.e. a link that loads a linkbase file. A linkbase can be loaded in two ways depending on the value of the *actuate* attribute. If the value is "onLoad", the linkbase is loaded and its links extracted as soon as the starting resource is loaded. An example of a linkbase is given in Figure 17.

```
<basesloaded xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="extended" >
    <startrsrc xlink:type="locator" xlink:label="spec" xlink:href="spec.xml" />
    <linkbase xlink:type="locator" xlink:label="linkbase" xlink:href="linkbase.xml" />
    <load xlink:type="arc" xlink:from="spec" xlink:to="linkbase"
    xlink:arcrole="http://www.w3.org/1999/xlink/properties/linkbase"
    xlink:actuate="onLoad" />
</basesloaded>
```

Figure 17. An example of a linkbase

3.4 XPointer

XML Pointer Language (XPointer) allows portions of XML documents to be identified and addressed in terms of their placement in the XML document. XPointer provides a way to refer to a certain point or range of selections inside an XML document. This allows referencing of any portion of an XML document without having to modify that document. XPointers are included within the portion identifier part of a URI attribute value. They vary from points to complex portions and can even be distributed over the documents.

The XPointer mechanism is useful, for example when composing new document assemblies from existing parts. XPointer can go deep inside to portions of an XML document. Where XPointer identifies sets of locations there, XLink connects and describes them.

3.5 Summary

XLink's multiple links can connect an arbitrary number of resources and define traversal rules between the resources. Rules determine which participating resources are starting and which are ending resources. Multiple links can contain multiple links as well. It is possible to create and maintain several context dependent linkbases which can be associated with a set of XML documents.

XLink specification provides a means to enrich Web-based linking. It provides a test-bed for formalising, constructing, demonstrating, implementing and evaluating diverse linking structures. However, if XLink is to be widely accepted and used, native support in mainstream browsers is needed. At the moment XLink is used in customized and professional browsers and applications.

Section 4

Time-Sensitive Linking Structures

The purpose of this section is to introduce a model for time-sensitive linking structures, XLinkTime, which can be applied to professional Web applications where links between resources and/or portions of resources are functions of time. Two approaches, bottom-up and top-down, to construct time-sensitive linking structures are presented. These issues are discussed in Paper 3, Paper 4 and Paper 5.

4.1 Temporal Relations

The Compact Oxford English Dictionary gives the following definitions for time (AskOxford 2004):

- The indefinite continued progress of existence and events in the past, present and future, regarded as a whole.
- A point of time as measured in hours and minutes.
- The favourable or appropriate moment to do something.
- An indefinite period.
- A portion of time characterized by particular events or circumstances.
- A period regarded as characteristic of a particular stage of one's life.
- The length of time taken to complete an activity.
- An instance of something happening or being done.

These definitions mainly reflect the characteristic way of humans to understand time in terms of events. According to Giumale and Kahn (1993) time, as an abstract concept, means a space of time points reached from one another by *before* and *after* -like operators. Some essential features in the nature and in the structure of time have been reported by Schreiber (1994):

- absolute ordering (past, present, future)
- relative ordering (before, concurrent-with, after)
- finiteness and infinity
- openness and closure
- discreteness and continuity
- objectivity and subjectivity
- linearity and circularity, i.e. periods.

Time can be represented in terms of time primitives like durations, duration bounds, time points and pieces of time, i.e. time intervals. Duration is the absolute distance between two points in time specified in terms of years, months, weeks, days, hours, minutes and seconds. Duration boundaries are defined by an upper and lower boundary such as a minimum duration of 2 minutes and a maximum duration of 20 minutes. Time points are used to represent specific, instantaneous points along a timeline. Time intervals are a set of constraints between two points, a start and an end time.

When modelling time, there are two main traditions represented in the literature. One view of time is a set of points without duration (Van Benthem 1991), and the other is that intervals should be considered as temporal individuals (Allen 1983; Allen 1984; Allen and Kautz 1985; Allen 1991). Berztiss (2002) has pointed out that Allen does not define an interval by time points. For example, if given the intervals (t_1, t_2) and (t_2, t_3), the question "to which interval does the time point t_2 belong" is not an issue in Allen's interpretation. According to Allen's interpretation, an interval is an undefined basic concept the meaning of which derives from the relations in which it stands with other intervals -relations such as "overlaps", "contains", "comes before" etc.

In addition to Allen's 13 relationships between two time intervals, the definition for an interval in this thesis is complemented by duration, a quantitative part. Time interval X can be regarded as being a time object used in this thesis and it is defined by the two points, x_s and x_e, where $x_s \le x_e$ as $X = \{ t \mid x_s \le t \le x_e \}$ and it is denoted as [x_s, x_e]. In the expression [x_s, x_e], x_s and x_e denote the start point and end the point of the interval, respectively. The duration of an interval is $D = x_e - x_s$. An instant is defined an interval with zero-length duration in time.

The temporal relationships between two time objects X and Y and corresponding functions implemented with JavaScript are given in Table 4. These temporal functions are used in the proof-of-concept implementations. The functions include two time objects, startTime and endTime. When the application includes dates, the value of the date in the number of milliseconds is returned by the getTime method since 1st January 1970, and is defined as 00:00:00.

Table 4. Temporal relationships between two time objects $X = \{ t \mid x_s \le t \le x_e \}$ and $Y = \{ t \mid y_s \le t \le y_e \}$ according to Allen's relations and corresponding functions implemented with JavaScript

Relations	Interpretation of relations	Example	Inverse
X before Y	From end X to beginning Y. **function before(x,y)** { return (x.endTime.getTime()) < (y.startTime.getTime()); } **Inverse:** **function nbefore(x,y)** { return !before(x,y); }	XXX YYY	YYY XXX
X equals Y	From beginning X to beginning Y, and from end X to end Y. **function equal(x,y)** { return ((x.startTime.getTime()) == (y.startTime.getTime())) && ((x.endTime.getTime()) == (y.endTime.getTime())); } **Inverse:** **function nequal(x,y)** { return !equal(x,y); } Table 4. Continues …	XXX YYY	YYY XXX

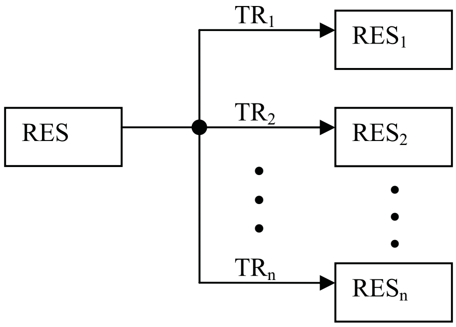
X meets Y	From end X to beginning Y. **function meets(x,y)** { return (x.endTime.getTime()) == (y.startTime.getTime()); } **Inverse:** **function nmeets(x,y)** { return !meets(x,y); }	XXXYYY	YYYXXX
X overlaps Y	Right overlaps: From beginning X to beginning Y, and from end X to end Y. Left overlaps: From beginning Y to beginning X, and from end Y to end X. Covering overlap: From beginning X to beginning Y, and from end Y to end X. Covering overlap: From beginning Y to beginning X, and from end X to end Y. **function overlaps(x,y)** { return ((x.startTime.getTime()) <= (y.endTime.getTime())) && ((x.endTime.getTime()) >= (y.startTime.getTime())); } **Inverse:** **function noverlaps(x,y)** { return !overlaps(x,y); }	Right overlaps XXXX YYYY Left overlaps XXXX YYYY Covering overlap XXXXXXX YYY Covering overlap XXX YYYYYYY	YYYY XXXX YYYY XXXX YYYYYYY XXX YYY XXXXXXX
X during Y	From beginning Y to beginning X, and from end X to end Y. **function during(x,y)** { return ((x.startTime.getTime()) >= (y.startTime.getTime())) && ((x.endTime.getTime()) <= (y.endTime.getTime())); } **Inverse:** **function nduring(x,y)** { return !during(x,y); }	XXX YYYYYYY	YYY XXXXXXX
X starts with Y	From beginning X to beginning Y. **function starts(x,y)** { return (x.startTime.getTime()) == (y.startTime.getTime()); } **Inverse:** **function nstarts(x,y)** { return !starts(x,y); }	XXX YYYYYY	YYY XXXXXX
X ends with Y	From end X to end Y **function ends(x,y)** { return (x.endTime.getTime()) == (y.endTime.getTime()); } **Inverse:** **function nfinishes(x,y)** { return !finishes(x,y); }	XXX YYYYY	YYY XXXXX

Time-dependent tasks, such as time-based project management or the maintenance instruction procedures of a technical object, are functions of time-intervals. These functions can be regarded as temporal rules related to certain tasks and situations. In this thesis Allen's relationships between two time intervals is used to describe temporal rules and express them in natural language. Time in this thesis is regarded as being a collection of intervals with defined start and end points.

4.2 An XLinkTime model for time-sensitive linking structures

In time-sensitive Web applications it is possible to identify three main categories. Firstly, a resource can be a function of time, such as news resources describing an accident. Secondly, resources and links between them can be functions of time. Versioning a software document is one example of this. Thirdly, only links between resources can be functions of time and the resources remain stable. The third category is the focus of this thesis. The basic assumption used in this thesis is that temporal rules can be identified in advance and they are used in task and situation specific circumstances. An example is the maintenance operations of a technical object where maintenance corresponds with the situation and operations correspond to the tasks.

An XLinkTime model for time-sensitive linking structures consists of resources and/or portions of resources and time-dependent links between them. Figure 18 shows an XLinkTime model (top) and an example of realization of XLinkTime by means of extending XLink by defining a timerule namespace, xmlns:timerule, with attributes timerule:start, timerule:end, timerule:status and timerule:title (bottom). XLinkTime can also include nested structures as illustrated in Figure 13.



XLink is extended by defining a timerule namespace *xmlns:timerule* with attributes *timerule:start*, *timerule:end*, *timerule:title* and *timerule:status*.

An XLinkTime model consists of temporal rules TR_n, (n = 1, 2, …) and related resources RES_n (n = 1, 2, …). The *timerule:start* and the *timerule:end* attributes can be used to specify the whole time interval $T_{whole} = [T_{start}, T_{end}]$ which consists of all temporal rules TR_n. The *timerule:start* and the *timerule:end* attributes is used with an attribute in the XLink namespace called *type* with a value of "extended" i.e. xlink:type="extended". An example is:

```
<project xmlns:timerule="http://www.tut.fi/amc/2003/05/timerule"
xmlns:xlink="http://www.w3.org/1999/xlink" xlink:label="Label" xlink:type="extended"
timerule:start="StartMonth, StartDay StartYear" timerule:end="EndMonth, EndDay
EndYear">….</project>
```

Inside xlink:type="extended" there can be n time-sensitive links with related resources. They are defined by an attribute in the XLink namespace called *type* with a value of "locator" i.e. "xlink:type="locator". An example is:

```
<remote xlink:type="locator" xlink:label="Label" xlink:title="Title of a link"
xlink:href="file.html" timerule:start="StartMonth, StartDay StartYear"
timerule:end="StartMonth, StartDay StartYear" />
```

With a *timerule:status* attribute one resource, for example an XML document with a pre-defined life cycle can be split into parts according to the sub life cycles of the document. In this thesis "new", "in work", "for approval", "released" and "obsolete" are used as examples for status names. Basically, the context defines the names of the statuses.

A *timerule:title* attribute has the same idea as xlink:title. It can be used for human-readable purposes. For example, with timerule:title="8 hours maintenance" the text string "8 hours maintenance" will be shown to a user, and the XML document can include a temporal rule defined by timerule:start="0h" and timerule:end="8h".

An xlink:type="extended" can include timerule:start and timerule:end attributes. An xlink:type="locator" can include timerule:start, timerule:end, timerule:title and timerule:status attributes.

Figure 18. Top: An XLinkTime model with temporal rules TR_n, (n = 1, 2, …) and related resources RES_n (n = 1, 2, …). In XLinkTime each participating link is activated by a user when the appropriate temporal rule is valid. Bottom: An example of the realization of XLinkTime by means of extending XLink's multi-ended link with timerule-namespace

4.3 From a Bottom-up to a Top-down Approach

The problem of describing time-sensitive linking structures can be approached in two ways: the top-down and bottom-up approach. These approaches are discussed in Paper 4. The bottom-up approach concentrates more on constructing time-sensitive linking rules on the content base level and is more dependent on the resources involved. The top-down approach could be applied to describe both application specific and general temporal rules. The focus of this thesis is on the bottom-up approach. However the idea of the top-down approach as an extension to bottom-up is briefly described here.

The top-down approach consists of two main phases: (a) modelling time in general by means of time primitives, and (b) identifying application specific temporal rules. Tools for building ontologies and XML Topics Maps (XTM) may be one practical means of implementing the top-down approach (ISO/IEC 13250; Park and Hunting 2003; XTM TopicMaps.Org. 2001). In a time-ontology approach, the ontology is an explicit artifact, distinct from the contents related to an application (Zhou and Fikes 2002). The design of the ontology can be separated from the content space. The same time ontology can be overlaid on different contents to provide different views to users according to the temporal rules under consideration. One advantage of an ontology-driven approach is its maintainability. Figure 19 shows a case where the time concept space is separated from the document space. An example of this kind of approach could be multi-project management environments in an enterprise that carries out and is responsible for many projects at the same time. Projects can have general temporal rules and each project can have temporal rules of its own.

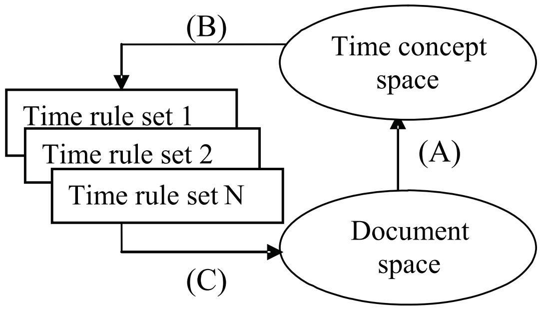


Figure 19. (A) Separating time concept space and document space. (B) Different time rule sets can be generated from time concept space and (C) applied to appropriate subset of a document space. Time concept space can consist of general and application specific temporal rules

4.4 Summary

A model for time-sensitive linking structures, XLinkTime, was introduced in this section. Realization of XLinkTime model based on XLink Recommedation was described. XLink is extended by defining a timerule namespace xmlns:timerule with attributes timerule:start, timerule:end, timerule:title and timerule:status. Temporal functions according to Allen's relations implemented with JavaScript were given. Bottom-up and top-down approaches were discussed as two means for constructing time-sensitive linking structures.

Time-sensitive linking structures can be applied to applications where links are time-critical or where time is an essential viewpoint of an application. Time-based information services during cultural events, for instance, scheduled advertising on the Web, time-based assembling of training resources, flight schedules and link releases are examples of applications where time is an essential factor. Link release means transforming a link from private status i.e the link is available for example only on an organisation's intranet, to public status i.e. the link is available on Internet, at a certain time instant for a certain time interval. Examples of contents for link releases are invitations to submit tenders, job advertisements, press releases, and new product announcements. Time-based project management and composition of maintenance instruction procedures for a machine in maintenance situations are discussed in the next section as two professional application areas.

Section 5

Proof-of-Concept Implementations of XLinkTime

Hypertext and hypermedia systems have been applied to a variety of documentation applications. Distributed project management and technical documentation are two professional application areas of hypertext ever since the first commercial implementations for workstations and personal computers occurred (Barnett 1998; Cooke and Williams 1989; Craig 1992; Goldie 1997; Grønbæk et al. 1993; Grønbæk and Trigg 1999; McGuire et al. 1998; Wang and Rada 1998; Wang et al. 2001; Wei et al. 1996). These application domains are also the prime application areas of this thesis. Advanced linking and browsing models, that support contexts of use, started to develop in parallel with the Web and XML (Albers 2003; Boy 1997; Bry and Kraus 2002; DocZilla 2003; Price 1997; Rauch et al. 1997).

Augmenting time-based project management training by means of a dynamic project workflow chart and composition of technical maintenance instructions are two interesting application domains for time-sensitive linking structures. The aim of this section is to describe two proof-of-concept implementations of the XLinkTime and to evaluate the XLinkTime concept, model, method and instantiation. Proof-of-concept implementations are described in Paper 5 and Paper 6. Background discussions are presented in Paper 1 and in Paper 2.

5.1 Proof-of-Concept Implementation Issues

Several good textbooks are available in the area of Web application development (Comer 2001; Coulouris et al. 2001; Deitel et al. 2003; Ford 2004; Goldfarb and Prescod 1998). The books primarily focus on networking, markup and scripting languages and technologies for creating dynamic Web sites.

Client-server architecture is the basic architecture for professional Web applications (Elmasri and Navathe 1994) (Figure 20) in this thesis. The client provides the user interface for Web-based applications. The Web server is responsible for interaction between a client and resources. The Web server accepts a user request for data from a client, retrieves the data and then responds to the client request. In this thesis XML document repositories correspond to resources and/or portions of resources in the client-server architecture.

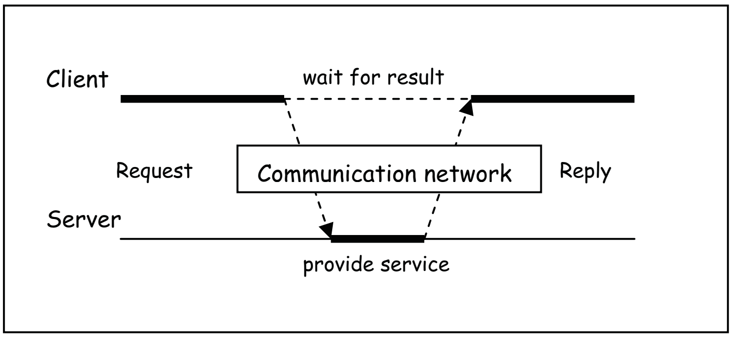


Figure 20. Client-server architecture

Basically three approaches to proof-of-concept implementation of time-sensitive linking structures can be identified with current technology as discussed in Paper 5. The approaches are illustrated in Figure 21. They usually differ only by the deployment of components. It is clear that the situation may change rapidly because of the development of Web technologies, and that other approaches may also exist. The approach to be considered is a matter of application domain requirement specifications, as well. The three approaches identified in this work are as follows:

1. The specialised service at the server side transforms the time-sensitive data to HTML and there are no special needs for the client. Time rules can be kept hidden in the service, or they can be brought to the client and they will be accessible by JavaScript. The service has an interface that uses the Extensible Stylesheet Language Transformations (XSLT) or similar technology for rendering the view. The approach is suited best to public Internet applications and can be used with legacy HTML-browsers.
2. The client transforms the data. This reduces the load at the server side and clients are responsible for generating the views. Contemporary browsers support XSLT transformations. This approach is suited to public Internet application, but better to extranet and intranet applications because of more versatile browser support. XML User Interface Language (XUL) is a markup language for describing user interfaces and it provides more advanced ways to generate user interfaces than HTML.
3. The server is a central repository for time rules. Client browsers are more sophisticated and have integrated support for time-sensitive linking. The approach is best suited to extranet and intranet applications.

There are several improvements when moving from the first approach to the third one. The server will only be used for data delivery, not for rendering. The time-sensitivity changes from emulation to integration. When time-sensitivity is integrated to a client, it can provide more control for the user, such as related menus and pop-up windows.

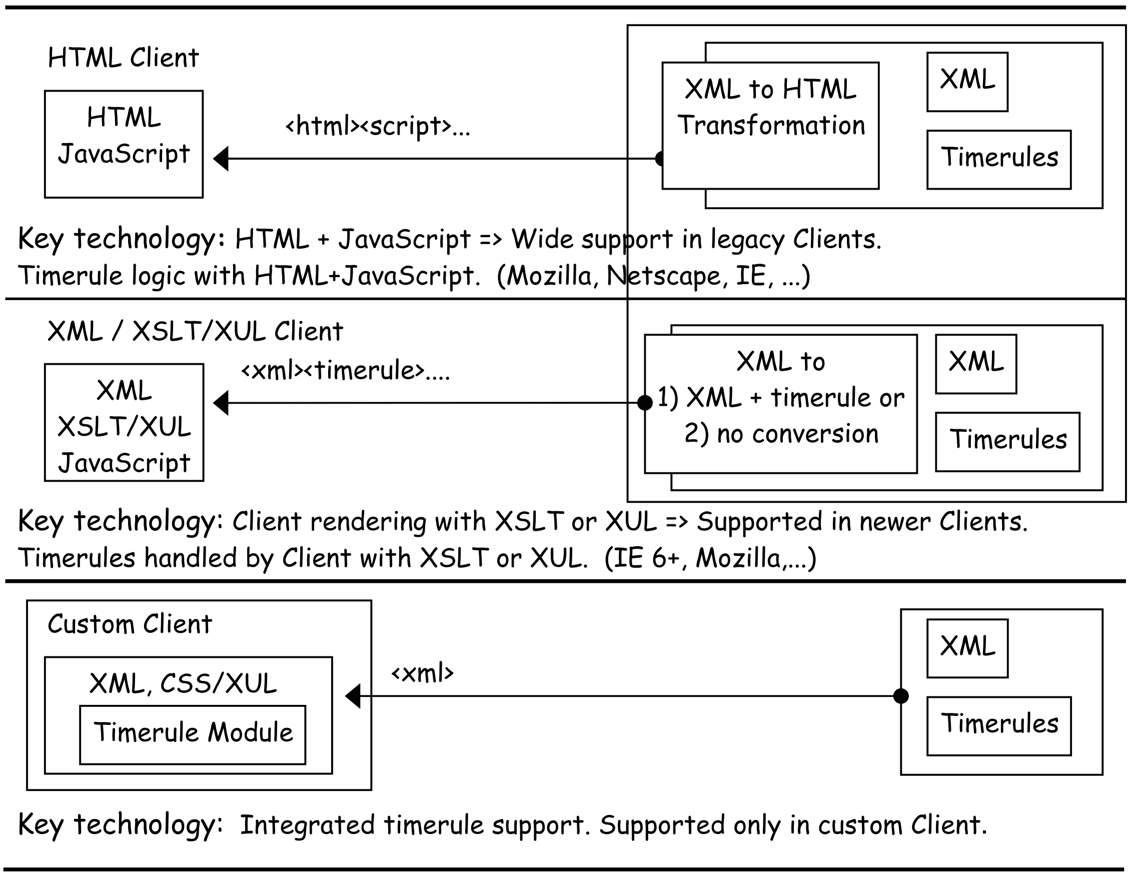


Figure 21. Three approaches for constructing proof-of-concept implementations of time-sensitive linking structures. Requirements for browsers increase from the top-down and for servers from the bottom-up

In the case of this research the second approach was best suited to the proof-of-concept implementations because of the support for XML document repositories, XLink, Mozilla, JavaScript and XUL technologies (J2EE 2004; Mozilla 2005; XUL 2005).

5.2 Augmenting Time-based Project Management Training by Means of a Dynamic Workflow Chart

Distributed project management systems on the Web have an important role in project execution between different organisations, particularly when inter- and intra-organizational projects involve geographically dispersed teams in different time-zones. Such systems provide a virtual place and time independent project workspace for communication, information interchange, document management and monitoring the progress of the project. Within access rights, the virtual project workspace and tools are available to all project partners. They can work with a standard Web browser regardless of their geographical location and time. Distributed project management includes three main tasks: communication between project parties, monitoring project execution and knowledge management during the project's whole life cycle (Hameri 1997; Kronodoc 2004; Liebowitz and Megbolugbe 2003; Norton 2000).

A layer model of distributed project management integrates key disciplines for designing, executing, analyzing and understanding document-driven processes in project operations (Hameri and Nihtilä 1997; Hameri and Nihtilä 1998; Pillai et al. 2002). The model usually consists of four main layers: a document management layer, a communication layer, a project operations layer and business performance layer. The document management layer constitutes a repository for project deliverables with standard document management tools for document lifecycle management, document classification and on-line configuration of the project workspace. The communication layer implements tools for keeping project members informed about what other members are doing and thus increases project transparency and information transfer. The business performance layer implements tools for document process across the whole project workspace according to a company's strategy. The project operations layer also provides support for multi-project management. One of the essential design issues of distributed project management systems for document driven business performance has been the reduction of project lead-times (Hameri and Heikkilä 2002).

The main object of training time-based project management in enterprises is to create best practices for project execution. As a consequence project lead-times should decrease and this brings economical benefits with increased competitiveness. Life-cycles of documents is the core of a time-based project management approach. Life-cycles of documents and related statuses can be defined in the system after workflow and document analysis. Examples of a document's life-cycle statuses are *new*, *in work*, *for approval*, *released* and *obsolete*. For example, a group of documents can be in *for approval* status two weeks starting from May, 15 2005 and ending May, 31 2005 and then move to the *released* status.

Before implementing a time-based project management system, an organization has to analyze its project management process with related teams, tasks, project deliverables and documents. In the document-driven analysis, the organization defines document classification, document users and user groups, contexts and life cycles of documents, a structure of project portfolios and finally, as a consequence, a step-by-step or phase-by-phase workflow chart for the whole project.

An organization specific, project workflow chart is an essential tool for training project teams, members and managers to understand their roles, tasks and responsibilities in a project. The project workflow chart makes it easier for them to piece together their roles during the whole project. This may help project teams to adopt a new, forthcoming project management system and new working habits. It may add commitment and minimize possibly existing resistance to change.

However, training project progress, related operations and deliverables with a static workflow chart against the project schedule pose problems. Figure 22 presents a vision for augmenting a Web-based project management system with a top level dynamic project workflow chart. The concept "dynamic" in this context means that the views in the workflow chart depend on the temporal rules related to project tasks, project documents and their life-cycles. Temporal rules can be embedded in XLink linking elements inside XML documents or they may even be stored in a separate linkbase file. In multi-project environments temporal rules common to all projects and project specific sets of rules could be described by means of time ontology or a topic map.

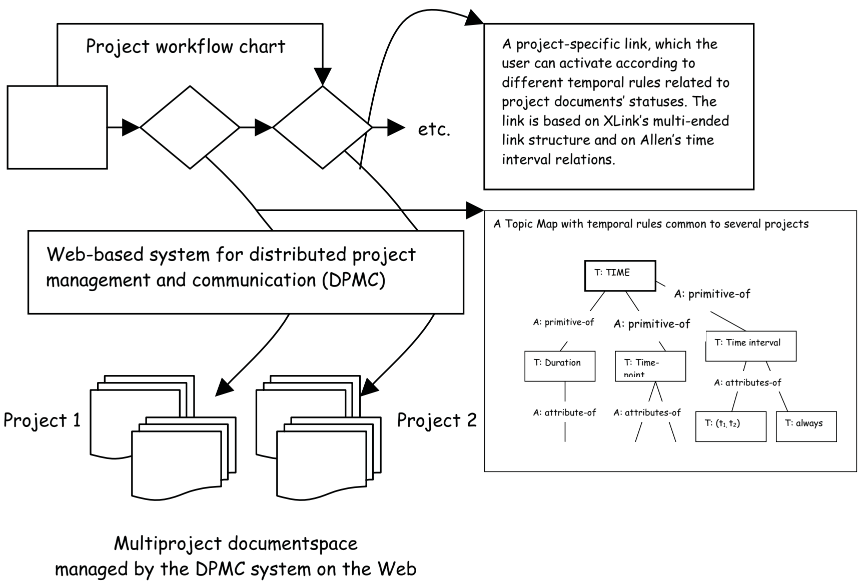


Figure 22. A vision for a dynamic project workflow chart with time-sensitive linking and navigation support. The vision is independent of the DPMC system

We have demonstrated time-sensitive linking and navigation support for a dynamic project workflow chart. With time-sensitive linking structures it is possible to illustrate the temporal characteristics of a project's tasks and task related documents. The basic idea is described in Paper 4 and a-proof-of-concept implementation in Paper 5. The application area of our demonstration is a software engineering project. A software development process is the set of activities needed to transform users' and technical requirement specifications into a software system. We have applied the Rational Unified Process (RUP) as a framework for the software engineering process (Jacobson et al. 1999).

RUP is based around four sequential phases that constitutes a development cycle and produce for a software generation (Jacobson et al. 1999). The phases are inception, elaboration, construction and transition. In the inception phase a business case, project's scope and a vision of the end product are specified. Project activities and resources are planned in the elaboration phase. The construction phase realizes the software product. During transition phase the product is transitioned to its users. Each phase can also contain several iterations. Each phase has phase specific deliverables. Phase specific documents can have different document statuses and temporal rules. Document statuses are described in Table 5, and temporal rules related to document lifecycles are presented in Table 6.

Table 5. Examples of document statuses

Name	Time unit	Description
New	Instant, interval	A new document is created
In work	Interval	A document is under construction.
For approval	Interval	Authorities evaluate the document(s).
Released	Instant, interval	A document has been approved.
Obsolete	Instant, interval	A document becomes outdated. In some cases a new version will be created and its critical time intervals for validity defined.

Table 6. Examples of temporal rules related to document lifecycles

Name	Time unit	Description
Ready before	Instant	A document must be ready before a given time instant.
Ready right-before (R-before) (*)	Interval (short)	A document must be ready just before a given time interval.
Ready after	Instant	A document must be ready after a given time instant.
Ready right-after (R-after) (*)	Interval (short)	A document must be ready just after a given time interval.
Ready equal	Instant	A document must be ready at a given time instant.
Overlap	Interval	Two documents or document groups share the same status during certain time interval.
Within/during	Interval	Documents with a certain status totally inside a given interval are chosen. In ”overlap” the status of a document may also exceed the given interval.
Start	Instant	Documents which start in a given time instant.
Finish	Instant	Documents which finish in a given time instant.

(*) For example, if the temporal rule in R-before is 20.8.2005 - 22.8.2005, the resulting set of documents are those which end in 20.8.2003 and if the R-After is chosen, the resulting set of documents are those which start 22.8.2003. The difference between the Before/After –temporal rules with R-Before/R-After is that in R-Before/R-After documents are stuck at the edge of the given interval. The Before-rule also selects those documents which have ended long before the given time interval.

An example of a view is given in Figure 23 and an example of the syntax in Figure 24. When the user activates a certain block in the flowchart, he/she can give a temporal rule and/or a document status to retrieve appropriate documents or he/she can navigate in a flowchart according to different temporal views. A specific search command can, for instance, be, “give me all documents that must have ‘for approval’ –status within one week” or “give me all documents that I am responsible for and that must have ‘for approval’ –status within one week”.

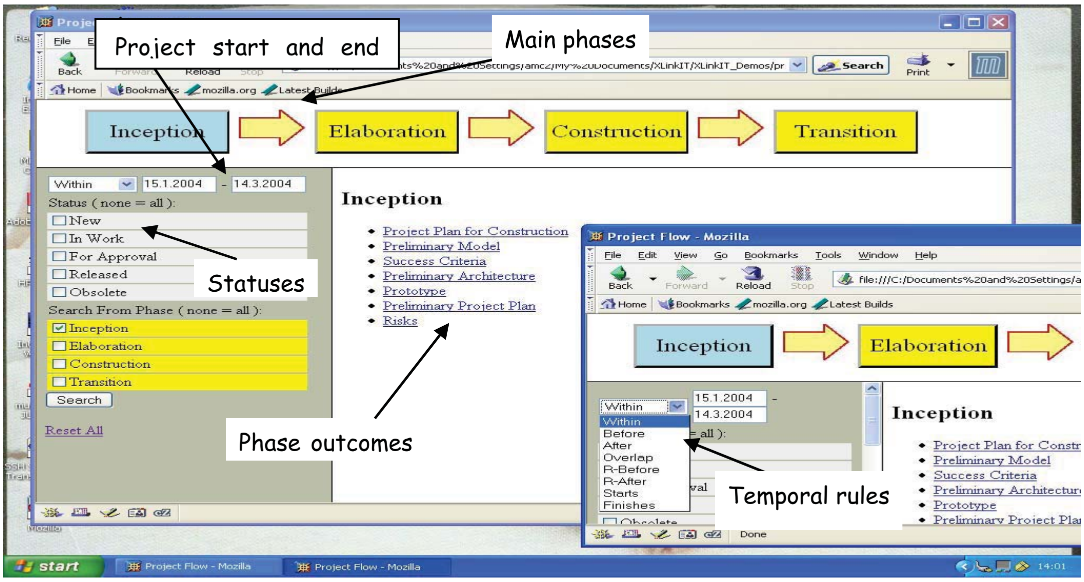


Figure 23. A dynamic project workflow chart with time-sensitive linking and navigation support

```
<project
    xmlns:timerule=”http://www.tut.fi/amc/2003/05/ti
    merule”
    xmlns:xlink=”http://www.w3.org/1999/xlink”
    xlink:type="extended"  ....>

  <local xlink:type="resource"
  xlink:title="Inception” xlink:label="Phase 1"/>

  <remote xlink:type="locator"
  xlink:label="Inception” xlink:title="Creating
  Preliminary Models”
  xlink:href="inception/premodel.html"
       timerule:status="new"
       timerule:start="Jan, 15 2006”
       timerule:end="Jan, 26 2006" />

  <binding xlink:type=”arc”
         xlink:from=”Phase 1”
         xlink:to=”Inception”/>

</project>
```

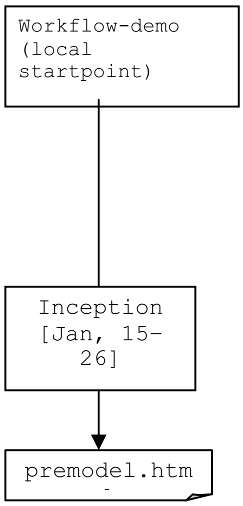


Figure 24. An example of a temporal rule defined by the timerule:status, timerule:start and timerule:end attributes. The rule is embedded in the XLink linking element

In our example the inception phase consists of six sub-phases: creating a preliminary model, designing preliminary architecture, constructing a prototype, analyzing risks, preparing a preliminary project plan and defining success criteria. As the outcome from each sub-phase there will be one document. Each document has time status development of its own related to other phases and to the whole schedule of the project. Syntax of the sub-phase “Creating a preliminary model” is given in Figure 25. Other sub-phases are formalized correspondingly.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="commonstyle.xml" ?>

<project xmlns:timerule="http://www.tut.fi/amc/2003/05/timerule"
xmlns:xlink="http://www.w3.org/1999/xlink" xlink:label="Inception" xlink:type="extended"
timerule:start="Jan, 15 2004" timerule:end="Mar, 14 2004">
        <!-- local -->
        <local xlink:type="resource" xlink:title="Inception" xlink:label="Phase 1"/>

    <!-- locators for req.doc-->
    <remote xlink:type="locator" xlink:label="Inception" xlink:title="Creating a
    Preliminary Models" xlink:href="inception/premodel.html"
    timerule:status="new" timerule:start="Jan, 15 2004" timerule:end="Jan,  26 2004" />
    <remote xlink:type="locator" xlink:label="Inception" xlink:title="Creating a
    Preliminary Models" xlink:href="inception/premodel.html"
    timerule:status="inWork" timerule:start="Jan, 27 2004" timerule:end="Feb, 12 2004" />
    <remote xlink:type="locator" xlink:label="Inception" xlink:title="Creating a
    Preliminary Models" xlink:href="inception/premodel.html"
    timerule:status="forApproval" timerule:start="Feb, 13 2004" timerule:end="Feb, 15
    2004" />
    <remote xlink:type="locator" xlink:label="Inception" xlink:title="Creating a
    Preliminary Models" xlink:href="inception/premodel.html"
    timerule:status="released" timerule:start="Feb, 16 2004" timerule:end="Jun, 1 2004" />
    <remote xlink:type="locator" xlink:label="Inception" xlink:title="Creating a
    Preliminary Models" xlink:href="inception/premodel.html"
    timerule:status="obsolete" timerule:start="Jun, 2 2004" timerule:end="Jan, 15 2005" />
…
<!-- arcs -->
        <binding xlink:type="arc" xlink:from="Phase 1" xlink:to="Inception" />

</project>
```

Figure 25. Syntax of the Inception phase

The demonstration runs on XUL-supported (XML User Interface Language) browsers, like Mozilla. The XML User Interface Language is a markup language for describing user interfaces and at the moment it provides more advanced ways of generating user interfaces than HTML. For example, search results in the demonstration are shown by means of XUL elements. In general, the Mozilla environment was chosen because of the more convenient Java programming techniques. The proof-of-concept implementation itself contains 15 files which are described in Table 7.

Table 7. Descriptions of the files related to the dynamic workflow chart demonstration

File name	File description
allen.js	The script contains Allen's relationships between two time intervals implemented with JavaScript.
xlinktime.js	The script contains the XLinkTime functionalities implemented with JavaScript. The script includes three main parts. (1) The first part contains four classes. The first class represents a time interval. The second, XLinkTime class, contains all XLink links found within one document. The third, the locator class, defines href-attributes and their time-relevancies. The fourth, the resource class, defines locators bound with arcs. (2) The second part of the script contains the function that creates an interval object from two date strings which must be in the format "dd.mm.yyyy". (3) The third part in the script is responsible for identification and returning an array of locators that meet the specified time context - for example, all locators that overlap with 10.01.2005 – 11.01.2005.
common.js	The script returns a given number as a double digit.
inception.xml	The file contains XLinkTime-links with attributes *timerule:status, timerule:start and timerule:end,* and locators for the Inception subphases.
elaboration.xml	The file contains XLinkTime-links with attributes *timerule:status, timerule:start and timerule:end,* and locators for the Elaboration subphases.
construction.xml	The file contains XLinkTime-links with attributes *timerule:status, timerule:start and timerule:end,* and locators for the Construction subphases.
transition.xml	The file contains XLinkTime-links with attributes *timerule:status, timerule:start and timerule:end,* and locators for the Transition subphases.
commonstyle.xml	The file parses the links for the phase files (inception.xml, elaboration.xml, construction.xml, transition.xml) and registers itself to search engine.
errorInternal.html	The file is a placeholder for possible internal errors.
index.html	The topmost file which divides the view on the screen into frames.
lower.html	The file divides the lower portion of the view on the screen into navigation and content frames.
upper.html	The file produces the four rectangles on the top frame.
main.html	The file starts up the demonstration.
navitime.html	The file is responsible of the navigational logic.
searchresult.xul	The file shows the search results in a list.

5.3 Time-Sensitive Composition of Technical Maintenance Instructions

An example of time-sensitive tasks related to technical documentation is carrying out the maintenance operations of a technical machine according to its maintenance schedule. Time contexts are defined by the service intervals of the maintenance schedule. Each service interval has its own maintenance instruction procedure. The second proof-of-concept implementation of time-sensitive linking structures of this thesis focuses on composing the maintenance instruction procedures of a rock grinder with related service intervals as time contexts (Paper 6). The rock grinder instruction manual has three textual hierarchy levels. Our demonstration focused on the third level:

Maintenance instruction →
Maintenance →
How to use the maintenance schedules

The grinder has seven service intervals: 8-hour maintenance, 50-hour maintenance, 200-hour maintenance, 400-hour maintenance, 800-hour maintenance, 1200-hour maintenance and 2400-hour maintenance. In this case, each interval deals solely with those items that are relevant to it but also encompasses the previous service items. For example, a 200-hour maintenance includes 8-hour + 50-hour + 200-hour services (Figure 26).

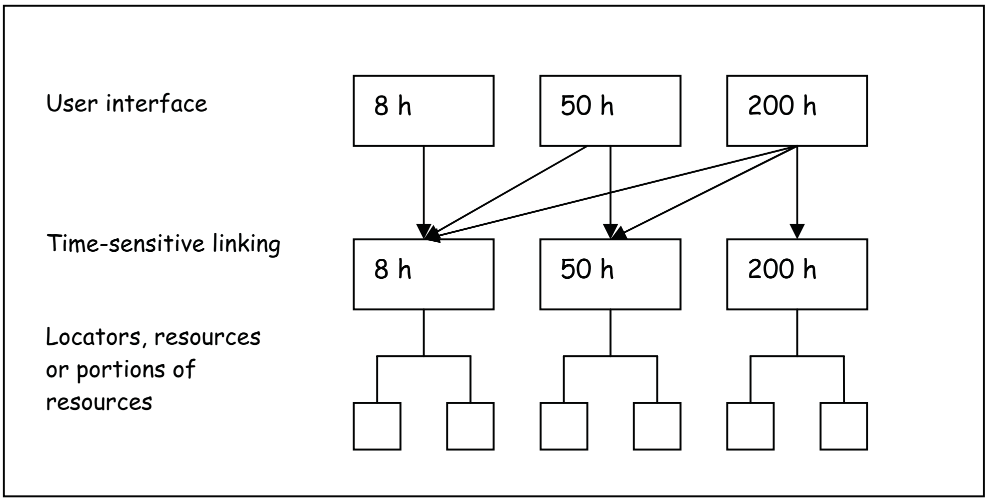


Figure 26. The 200-hour maintenance includes 8-hour + 50-hour + 200-hour services

The service interval instructions consist of between one and seven items to be taken into account. Each item includes from one to three tasks. The instructions are short texts, like check lists containing technical illustrations and some Note! remarks. Several time-sensitive contexts related to the service intervals of a rock grinder or a technical object in general can be identified, where the maintenance crew needs instructions. Examples of time contexts are discussed in Paper 6.

The demonstration contains five files which are described in Table 8. The demonstration supports external linkbase files. Portions of resources are identified by means of the XPointer language. Service intervals are defined by timerule:start and timerule:end attribute. A user selects a linkbase from a linkbase frame. Linkbase documents are loaded and a tree of documents is generated. The tree view consists of nodes that group maintenance functions according to the service intervals. The tree view consists of nodes that group every service interval by time, i.e. by 8h / 50h / 200h / 400h /

800h / 1200h / 2400h have their own nodes. The user can choose between different service intervals and related tasks according to the time condition which is at hand.

Table 8. Descriptions of the files related to the maintenance schedules demonstration

File name	File describtion
xlink.js	The script builds up the data model from XLinkTime links.
index.html	The file is the entry point to the demonstration and also splits the view into frames.
situations.xml	The file produces a tree-structure on the leftmost frame (see Paper 6, Figure 6). The contents are read from the simgrinder.xml file that has been loaded by the baseloader.xul.
simgrinder.xml	The linkbase file contains XLinkTime links to portions of the SIM Grinder manual. The portions are identified by means of the XPointer language.
baseloader.xul	The file loads the external linkbase, in this demonstration titled simgrinder.xml that contains XLinkTime links.

Syntax for the 50 h service is given as an example in the Figure 27. The syntax related to other service intervals was created correspondingly.

```
<?xml version="1.0"?>

<links  xmlns:timerule="http://www.tut.fi/amc/2003/05/timerule"
        xmlns:xlink="http://www.w3.org/1999/xlink"
         xlink:type="extended">

        <local xlink:type="resource" xlink:title="Maintenance schedules"
        xlink:label="ms"/>

        <!-- Maintenance Schedules -->

<!-- 50 h -->
        <remote xlink:type="locator" timerule:start="0 h" timerule:end="50 h"
        timerule:title="50 h"
        xlink:label="ms" xlink:title="Oil level"
        xlink:href="http://www.doczilla.com/demos/demos/simgrinder/manual/sim.xml#xpointer
        (/1/7/3/10/4/1)" />

        <remote xlink:type="locator" timerule:start="0 h" timerule:end="50 h"
        timerule:title="50 h" xlink:label="ms" xlink:title="Brake linings"

        xlink:href="http://www.doczilla.com/demos/demos/simgrinder/manual/sim.xml#xpointer
        (/1/7/3/10/4/3)" />

        <remote xlink:type="locator" timerule:start="0 h" timerule:end="50 h"
        timerule:title="50 h"
        xlink:label="ms" xlink:title="Counter knife and grinder shaft"
        xlink:href="http://www.doczilla.com/demos/demos/simgrinder/manual/sim.xml#xpointer
        (/1/7/3/10/4/5)" />

        <remote xlink:type="locator" timerule:start="0 h" timerule:end="50 h"
        timerule:title="50 h"
        xlink:label="ms" xlink:title="Hydraulic oil level"
        xlink:href="http://www.doczilla.com/demos/demos/simgrinder/manual/sim.xml#xpointer
        (/1/7/3/10/4/7)" />
```

```
    <remote xlink:type="locator" timerule:start="0 h" timerule:end="50 h"
    timerule:title="50 h"
    xlink:label="ms" xlink:title="The engine connection cables"
    xlink:href="http://www.doczilla.com/demos/demos/simgrinder/manual/sim.xml#xpointer
    (/1/7/3/10/4/9)" />

    <remote xlink:type="locator" timerule:start="0 h" timerule:end="50 h"
    timerule:title="50 h"
    xlink:label="ms" xlink:title="Batteries"
    xlink:href="http://www.doczilla.com/demos/demos/simgrinder/manual/sim.xml#xpointer
    (/1/7/3/10/4/11)" />

    <remote xlink:type="locator" timerule:start="0 h" timerule:end="50 h"
    timerule:title="50 h"
    xlink:label="ms" xlink:title="Hydraulic components and connections"
    xlink:href="http://www.doczilla.com/demos/demos/simgrinder/manual/sim.xml#xpointer
    (/1/7/3/10/4/13)" />

<arc xlink:type="arc" xlink:from="ms" xlink:to="ms" />

</links>
```

Figure 27. Syntax for the 50 h service interval

5.4 Evaluation of XLinkTime

In constructive research the research problem can usually be formulated in one way or the other (Järvinen 2001):

- Can we build a certain innovation and how useful is the particular innovation?
- Which kind should the certain innovation be, and how should we build the particular innovation?

March and Smith (1995) and Järvinen (2001; 2004) discussed evaluation criteria of an innovation in the context of the constructive research method. March and Smith (1995) differentiate between two cases regarding whether the construct, model, method or instantiation already exists or is totally lacking. According to March and Smith (1995), when building the first of practically any set of constructs, model, method or instantiation is considered as research, which has utility for an important task provided by the artifact. Furthermore, March and Smith (1995) have written that in this case, the research contribution lies in the novelty of the artifact and in the conviction of the claims that it is effective. Actual performance evaluation is not required at this stage. Järvinen (2001; 2004) extended this approach by proposing that although the outcome itself has merit, the potential importance of the new construct, model, method and instantiation should be evaluated. Use of the construct, model, method and instantiation is emphasized in the constructive research method. According to Järvinen (2001; 2004), a researcher can ask

- Whether the new construct better differentiates the phenomenon to which this construct refers from other phenomena than any other potential construct?
- Whether the new construct better describes the phenomenon to which this construct refers than any other potential construct?
- What are the potential benefits of the new construct in use?

The evaluation framework of an innovation for constructive research method were introduced by March and Smith (1995) and it was extended by Järvinen (2001; 2004).

They presented evaluation criteria for the construct, model, method and instantiation as well as descriptions of the research outcomes (Table 9).

Table 9. Research outcomes, descriptions of research outcomes and evaluation criteria

Research outcome	Description	Evaluation criteria
Construct/concept	Constructs or concepts form the vocabulary of a domain. They form a conceptualization used to describe problems within the domain and to specify their solution. Constructs can be highly formalized or informal.	1. Ease of use 2. Exploitability
Model	A model is a set of propositions or statements expressing relationships among constructs. A model can play two roles. First, it can be viewed simply as a description, that is, as a representation of how things are (initial state). Secondly, the model can represent the target state. The target stage is not real, but desired, prescribed until the innovation is implemented, if ever.	3. Internal consistency 4. Maintenance
Method	A method is a set of steps (an algorithm or a guideline) used to perform a task. Methods are based on a set of underlying constructs (language) and a representation (model) of the solution space.	5. Operationality (the ability to perform the intended task or the ability of humans to effectively use the method if it is algorithmic) 6. Efficiency 7. Generality 8. Ease of use 9. Application domain 10. Maintenance
Instantiation	An instantiation is the realization of an artifact in its environment. Instantiations operationalize constructs, models and methods. Instantiations also demonstrate the feasibility and effectiveness of the models and methods they contain.	11. Efficiency 12. Effectiveness of the artifact 13. Artifact's impacts on the environment and its users 14. Emergent changes with positive and negative unanticipated outcomes 15. Maintenance

XLinkTime is evaluated to an appropriate extent according to the evaluation framework given in Table 9. The research outcomes are described first and then they are evaluated against the criteria. The evaluation is presented in Table 10.

Table 10. Evaluation of XLinkTime

Concept	The concepts of XLinkTime are (1) Web resources and/or portions of Web resources with associated temporal rules, (2) a time-sensitive link and (3) a time-sensitive linking structure.
Ease of use	All three concepts are understandable and easy to use among IT personnel. However project managers and end users especially may need further clarification with the second and the third concept. The first concept is understandable and easy to use among project managers and end users in the application domain under consideration.
Exploitability	In addition to the application domains demonstrated in this thesis, the concepts can be utilized in other areas as well where time-sensitive knowledge structures can be identified. The exploitability of XLinkTime-like time-sensitive linking structure would increase and be world-wide spread if it were included in the next generation linking language recommendation to be defined by the W3C.
Model	The XLinkTime model is a graphical representation which consists of a multi-ended link structure with temporal rules defining when the appropriate link should be activated. By means of Allen's relationships application specific temporal rules can be described. For realizing XLinkTime, XML Linking Language (XLink) is extended by defining a timerule namespace *xmlns:timerule* with four attributes *timerule:start*, *timerule:end*, *timerule:title* and *timerule:status*.
Internal consistency	Temporal rules must be logical and consistent in the time-sensitive linking structure.
Maintenance	There is no problem of maintaining the graphical presentation of the XLinkTime model. At the moment, extending XLink provides one way of realizing the XLinkTime model. However, time-sensitive linking structures can be adopted by any other XLink alike linking language, including multi-ended linking, link traversals and linkbases, on the future Web.
Method	The XLinkTime method connects resources and/or portions of resources with pre-defined temporal rules. The method consists of four main steps. 1. Identification of temporal rules of the application under consideration, related resources and/or portions of resources, users and/or user groups, tasks and situations. 2. Creation of graphs describing the time-sensitive linking structures consisting of temporal rules, resources and/or portions of resources, and descriptions of user groups, tasks and situations. 3. One example for realizing the XLinkTime is to extend XLink by defining a timerule namespace *xmlns:timerule* with attributes *timerule:start*, *timerule:end*, *timerule:title* and *timerule:status*.

	4. The attributes of the timerule namespace can be defined according to the application domain that shows the flexibility of the method. 5. Mapping the identified temporal rules into XML based syntax and creation of functions based on Allen's relations between time intervals. Functions corresponding to Allen's relations can be realized with JavaScript, for example.
Operationality	The XLinkTime method is designed to operate in professional Web-based applications. In this thesis, professional Web-based applications are applications, where the temporal rules can be identified and specified in the application design phase, where the amount of resources and the number of associated links are limited. The user's information needs are task and situation oriented and navigation directed.
Efficiency	The implementation requires the time analysis of the application and related documents.
Generality	The XLinkTime method is customizable. The attributes of the timerule namespace can be defined according to the application domain by means of a time analysis of the application.
Ease of use	The use of the XLinkTime method requires know-how on the application domains concerned, Allen's relations, and hypertext linking. The know-how needed for realization of the XLinkTime depends on the selected technologies.
Application domain	In this thesis the XLinkTime method has been applied to time-based project management and to composition of maintenance instructions from a technical manual. The method can be applied to all domains where time-sensitive knowledge structures can be identified.
Maintenance	Basically, there is no problem of maintaining the XLinkTime method itself. The problems may arise from the solutions selected for the realization, because of rapid changes and development of Web recommendations and technologies.
Instantiation	The instantiation of the XLinkTime's concept, model and method has been realized as two proof-of-concept implementations. Allen's relationships between two time intervals are implemented by means of JavaScript, XLinkTime by means of extending XLink, XLinkTime functionalities by means of JavaScript. User interface is designed by means of XML User Interface Language (XUL). Mozilla environment was chosen because of the XUL-support and more convenient Java programming techniques.
Efficiency/effectiveness of the artifact	In general XLinkTime model and method can be realized in at least two ways: (1) by bottom-up approach, i.e. by embedding time-sensitive links into XLink linking elements or by storing sets of time-sensitive links in the separate linkbase files and (2) by a top-down approach where temporal rules can be described by means of time ontology and they are separated from the resources. The selection and effectiveness of these two proposed solutions depends on the number of links and on the frequency of the

	links' updating. If the time-sensitive link structures are static and the number of links is limited, then the embedded solution is effective. If there are both static and dynamic link structures then a linkbase solution could be recommended. If most of the links and/or link structures are dynamic, the number of links is huge and different sets of links are associated with the same resource space, then a linkbase and/or time-ontology approach could be recommended.
Artifact's impacts on the designing environment and its end users	Browsers which support multi-ended links and linkbases defined in the XLink specification could be extended with a time-sensitive linking module. The module can provide time context-based authoring functionalities for application designers and value-added information searching and navigation services for end-users.
Maintenance	New versions of JavaScripts and XUL language would require updating of the proof-of-concept implementations. However, it is important to highlight that the main aim of the proof-of-concept implementations presented in this thesis is to illustrate the general idea of time-sensitive linking structures, not to give recommendations for technical solutions. Depending on the requirement specifications of an application domain, there can also be other interesting approaches to implement time-sensitive links. In the near future, for example, the next version of the XLink linking language, ontologies, integration of Resource Description Framework and Topic Maps or even customized browsers with support for time-sensitive applications can provide alternative technologies for implementation process.

5.5 Summary

Two proof-of-concept implementations of XLinkTime have been described in this section. The application domains were time-based project management and composition of technical instructions. The evaluation of XLinkTime has been presented in this section.

Time-sensitive linking structures can be realized by embedding time-sensitive links in XLink linking elements or by storing time-sensitive links in separate linkbase files. The XLinkTime approach is suitable for professional Web applications where temporal rules can be identified and defined in the application design phase and where temporal rules are task and situation dependent.

In this thesis XLink has been used for the realization of XLinkTime. However, time-sensitive linking structures can be adopted by any other XLink resembling linking language on the future Web. The explicit XML based syntax of XLinkTime is application dependent and implementation requires the time and resource analysis of the application concerned. Implementation also depends on the life-cycle of the application, maintenance requirements, user needs and delivery platforms.

As a general remark, more complex and versatile linking structures can be utilized in extranet and intranet environments because users' computing environments can be supported with consistent advanced browser technology.

Section 6

Conclusions

In this thesis concepts of hypermedia linking research in the early days, such as multiple linking, link traversals and external linkbases, are applied to the Web environment by means of the W3C Recommendations instead of ad hoc hypermedia installations. In this thesis link traversals are defined by temporal rules inside a multi-ended XLink linking element. A model for time-sensitive linking structures, XLinkTime, is introduced. Realization of the XLinkTIme model is defined and constructed according to a temporal analysis of the application. Time-sensitive links can be viewed in two ways. First, links can be shown to the user as groups of resources where grouping is done according to temporal rules. Secondly, the user can search resources or portions of resources according to a given time notion. The former represents a static linking view and the latter a dynamic linking view. In this thesis temporal relationships are related to situations and tasks in the application under consideration, and the amount of available resources and links between the resources is limited and a user's browsing behaviour is directed. The time context of a link defines appropriate resources and traversal rules between resources. The realization of time-sensitive linking structures is based on two W3C Recommendations: XML Linking Language and XML Pointer Language.

6.1 Results

The main goal of this thesis has been to develop a practical approach to construct time-sensitive linking structures for professional Web-based applications. In this thesis, professional Web-based applications are applications, where the most relevant usage contexts can be identified and specified in the application design phase, where the amount of resources and the number of associated links are limited. A user's information needs are task and situation oriented and navigation directed. Time-based project management and composition of technical maintenance instructions according to related time-contexts, as two examples of application domains, were discussed and proof-of-concept implementations constructed. In a wider sense, time-sensitive linking can be regarded as being one special case of context-based linking.

The research problem of the thesis was divided into four sub-problems. The sub-problems and the solutions presented in this thesis with references to the Papers and Sections of this thesis are summarized next.

1. Sub-problem
How can temporal relations be expressed?

Solution:
Allen's 13 relations between two time intervals were used as a basis for identifying and describing temporal rules. Allen's definition for an interval is complemented by a quantitative part in this thesis. Time rules must be logical and consistent in the time-sensitive linking structure. Rules are defined according to a temporal analysis of the application.

References to papers and sections of this thesis
Paper 3, Paper 4, Paper 5, Section 4

2. Sub-problem
What does time-sensitivity require from linking structures?

Solution:
Multiple linking, link traversals and linkbases are three basic structures which are used in modelling time-sensitive linking structures. Link traversal rules are dependent on temporal rules. An XLinkTime model was introduced.

References to papers and sections of this thesis
Paper 1, Paper 2, Paper 3, Paper 5, Section 3, Section 4

3. Sub-problem
What are the possibilities of existing Web-based linking languages with W3C Recommendation status to support time-sensitive linking structures?

Solution:
XLink supports multiple linking, link traversals and linkbases. XPointer supports addressing portions of resources. The linking language should also be extendable with a timerule namespace. XLink can be extended by defining a timerule namespace, which in this case is xmlns:timerule with attributes timerule:start, timerule:end, timerule:title and timerule:status. The attributes of the timerule namespace can be defined according to the application.

References to papers and sections of this thesis
Paper 5 and Section 4.

4. Sub-problem

How can proof-of-concept implementations of time-sensitive linking structures be realized?

Solution:
Two different approaches to realize XLinkTime model were discussed in this thesis: (1) a bottom-up approach, i.e. by embedding time-sensitive links in the XLink linking elements or by storing sets of time-sensitive links in the separate linkbase files and (2) a top-down approach where temporal rules can be described by means of time ontology and can be separated from the document space. The selection of a solution depends on the number of links and on the links' updating frequency. If the time-sensitive link structures are static and the number of links is limited, then the embedded solution is effective, easy to construct and maintain. If there are both static and dynamic link structures then a linkbase solution could be recommended. If most of the links and/or link structures are dynamic, the number of links is huge and different sets of links are associated with the same document space, then a linkbase with time-ontology approach could provide one solution.

Allen's relationships between two time intervals have been implemented by means of JavaScript, XLinkTime linking structures by means of extending XLink, XLinkTime functionalities by means of JavaScript, and user interfaces by means of XUL. The Mozilla environment was chosen because of XUL-support and more convenient Java programming techniques. The client-server architecture was used for proof-of-concept implementations. If we compare this architecture to the four stages in development of hypermedia architectures' (Figure 7) which demonstrate various component separations in hypermedia systems, we can conclude that:

- When time-sensitive linking structures are embedded in XLink linking elements in XML documents on the Web, the solution represents the first stage in development of hypermedia architectures.
- When time-sensitive linking structures are stored in linkbase file(s) which can be associated with XML documents the solution can be regarded as representing the second stage in development of hypermedia architectures.
- When the time ontology approach is applied to construct time-sensitive linking structures, the time concept space is separated from document space and the solution can be regarded as representing the third or fourth stage in the development of hypermedia architectures.

In this thesis it was shown that XLink's multiple linking, link traversal rules and linkbases with timerule namespace extensions can be used to realize time-sensitive linking structures in professional Web-based applications within the scope of the premises of this thesis.

References to papers and sections of this thesis
Paper 4, Paper 5, Paper 6 and Section 5.

The main observations of this thesis are summarized in Table 11.

Table 11. The main observations of this thesis

• Time is matter and an interesting dimension of context when a user accesses information from professional applications on the Web. • Time-sensitive application domains and sub-domains in one application area can be identified. • Time is related to professional tasks and situations. • As a result of integrating Allen's relations, multiple linking, link traversal rules and linkbases, an XLinkTime model is introduced. • By means of the XLinkTime model time-sensitive linking structures can be designed and constructed. • Two main approaches can be identified: a bottom-up approach and a top-down approach. The thesis focused on the bottom-up approach. However, the top-down approach was also discussed briefly. • Realization of the XLinkTime model can be done by extending XLink by a timerule namespace. Realization is demonstrated by means of proof-of-concept implementations. • The relations between the Web client-server architecture and the four stages in development of hypermedia architectures were identified. • Designing and constructing time-sensitive linking structures is one way to approach task and situation specific time contexts and related resources when a user accesses information from a professional application on the Web.

6.2 Discussion

When time-sensitive linking structures are implemented, several issues occur which need to be taken into account. Some essential questions and answers are presented in Table 12.

Table 12. Issues to be considered when implementing time-sensitive linking structures

Question	Answer
Is time a) "now", b) user decision, or c) related to some other entity?	It can be any of these and it should be stated somehow.
If time is "now", what is its interval?	Its interval is zero, or more precisely, the start time and the end time are equal and they have the same precision.
If time is "today", does it mean that the time interval is 0.00-23.59?	It could mean that, or it could mean a date with a zero interval.
If time is a user decision, how should time intervals be compared?	In professional Web-based applications time rules and related usage contexts can be predefined.
How can precise times be expressed in time sensitive linking?	Any precision can be used, for instance according to the ISO 8601 (ISO 8601).
Should the system prevent using links at the wrong time?	This is very important in professional applications.
How do time-sensitive link structures behave in applications concerning international organizations?	A mapping function between time zones is required in applications concerning international organizations. Time mapping across time zones can be implemented by means of Coordinated Universal Time (UTC) (W3C 1998a).

XLink has provided the basis for this research. However, time-sensitive linking structures can be adopted by any other similar XLink linking languages on the future Web. The position of XLinkTime in the fields of hypermedia research community, W3C and ISO is illustrated in Figure 28.

W3C and the open hypermedia research community together are developing the Web towards a world wide open hypermedia system. Research and development projects related to XLink give us valuable information which can cumulate into common knowledge space consisting of contributions from different research communities. This cumulative process is a logical phase in the evolution of hypertext linking research which helps us to proceed towards intellectual integration between the hypermedia research community and W3C community. The ongoing process also helps us to face one of the main still unsolved problems on the Web: management of links separately from contents. Perhaps help will be needed from the database research community before the open Web becomes a reality.

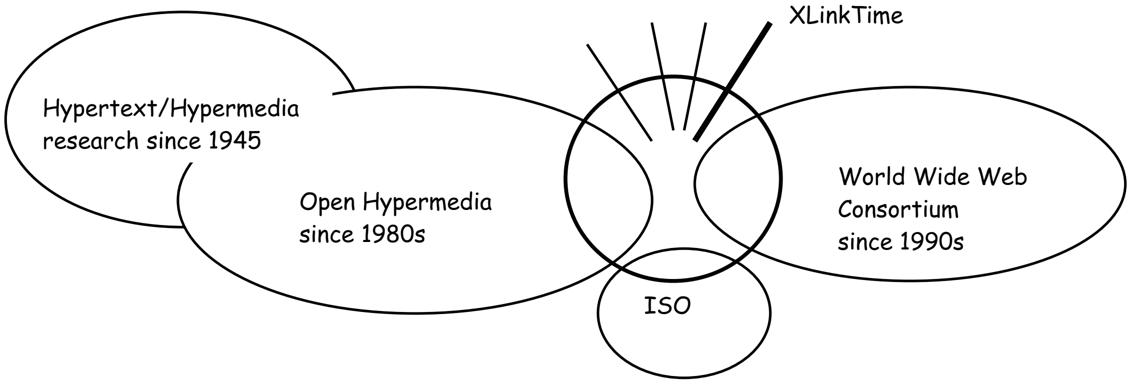


Figure 28. The position of the XLinkTime in the fields of hypermedia community, W3C and ISO

At the moment it seems that linking activities are scattered around the W3C. There are several working groups, and more integration is needed. For example, XHTML 2.0 (Extensible Hypertext Markup Language) does not include XLink's advanced linking features. W3C's Working Draft, "HLink: Link recognition for the XHTML", is one effort to provide the means for adopting advanced linking possibilities such as multiple linking defined in XLink specification to XHTML environments (W3C 2002).

The HTML Working Group is trying to coordinate linking activities with other groups both inside and outside of the W3C to resolve these issues. Hopefully together with the XLink Version 1.1 Working Group they can bring some integrative solutions and recommendations for linking on the Web (W3C 2005e).

One possible future vision for general level development of linking languages on the Web environments is illustrated in Figure 29. Nowadays constructing advanced linking functionalities needs a lot of customizing and browsers designed for special purposes. There are some of those, but could the future provide us with one advanced linking language with a modular structure going from simple to versatile link management, and what is very important, with a solid browser support? We will need that simply because contents and links are the core of the Web.

The broader development trend at the W3C is Semantic Web activity. The goal of Semantic Web Activity is to develop enabling standards and technologies, which are designed to help machines understand more information on the Web (W3C 2004d; W3C 2005b). When information on the Web becomes more machine-understandable computers can support more efficient searching, data integration, and automation of tasks, versatile linking and navigation structures. The fundamental idea of the Semantic Web is in providing a means to add specific information related to Web resources for automatic processing. The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries.

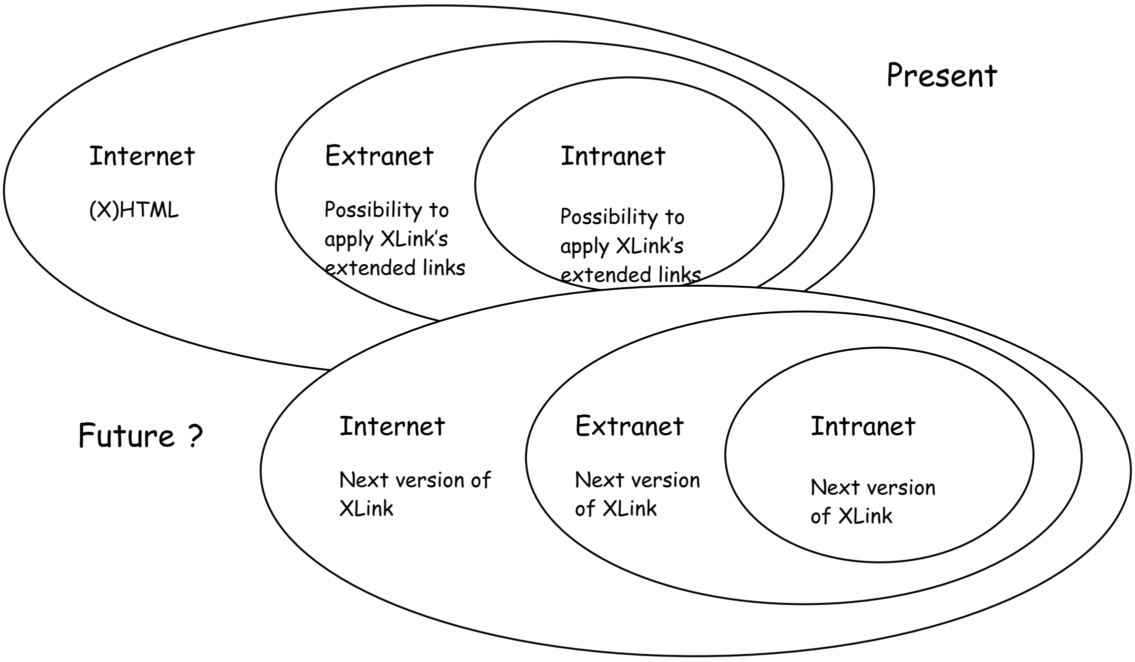


Figure 29. A future vision for general level development of a linking language on the Web

Semantic Web principles are implemented in layers of Web technologies and standards. The layers are presented in Figure 30 (Koivunen and Miller 2002) and the languages of the Semantic Web in Lassila et al. (2000). The Unicode layer makes sure that international characters sets are used and URI provides a means for identifying anything on the Semantic Web. The XML layer with linking, style and transformation, namespace and schema definitions forms the basis for Semantic Web definitions. With RDF and RDF Schema it is possible to describe resources with URI addresses and define vocabularies that can be referred to by URI addresses. The Ontology layer supports the evolution of vocabularies where relations between different concepts can be defined. The top layers, Logic, Proof and Trust, are under construction and they deal with inference functions.

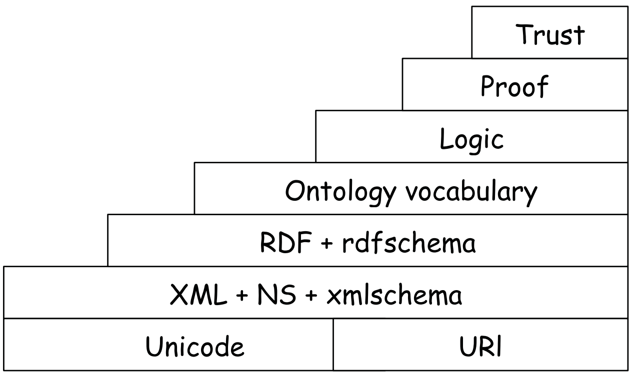


Figure 30. The Semantic Web layers

The XLink specification defines ways for XML documents to establish hyperlinks between resources. The Resource Description Framework specification defines a framework for the provision of machine-understandable information about Web resources (W3C 2004e) Both XLink and RDF provide a way of asserting relations between resources. RDF is primarily for describing resources and their relations, while XLink is primarily for specifying and traversing hyperlinks. However, the overlap between the two is sufficient that a mapping from XLink links to statements in an RDF model can be defined. Such a mapping allows XLink elements to be harvested as a source of RDF statements. The W3C's Note Harvesting RDF Statements from XLinks specifies such a mapping, so that links can be harvested and RDF statements generated (W3C 2000). The more interesting role of RDF is in its possibilities to describe meta information of complex link structures and linkbases. RDF Schema and ontology languages (W3C 2004f) can provide interesting tools for constructing both general and application domain specific link taxonomies.

Web Services (W3C 2004g; W3C 2004h) is a technology which seems to be taking the Web towards n-tier architecture. It may also be the missing link between the hypermedia research community and the W3C community to solve the main problems for managing links on the Web and creating a more open hypermedia-oriented Web (Komathy et al. 2003; Van van Ossenbruggen et al. 2002; Vivekanandan and De Roure 2002). Multiple open services aim to split up services provided at all layers in OHS architecture into components. Each component provides a functionally independent service. For example, the navigational component can be divided into a traverse link, start link, create anchor, delete anchor and end link services. A conceptual overview of the multiple open service idea is shown in Figure 31 (Wiil et al. 2001).

However, the task of integration between the OHS service and the WWW is complicated. Most Web integration efforts by the hypermedia research community are characterized by two main drawbacks: (a) hypermedia extensions and Web integrations are usually ad-hoc implementations which are created by non-standardized methodologies and (b) the OHS community has not made wide usage of standard Web technologies (Karousos et al. 2003; Nürnberg and Schraefel 2003). There are several reasons for these, the key argument being that most Web standards do not explicitly express a notion of links. Karousos et al. (2003) argue that Web Services, as an emerging recommendation for component-based software development, could well be a technology whose adoption by the OHS community would benefit both worlds. The OHS research community could use tools and technologies that are widely available. The Web community would benefit from richer linking and structuring facilities offered by OHS services.

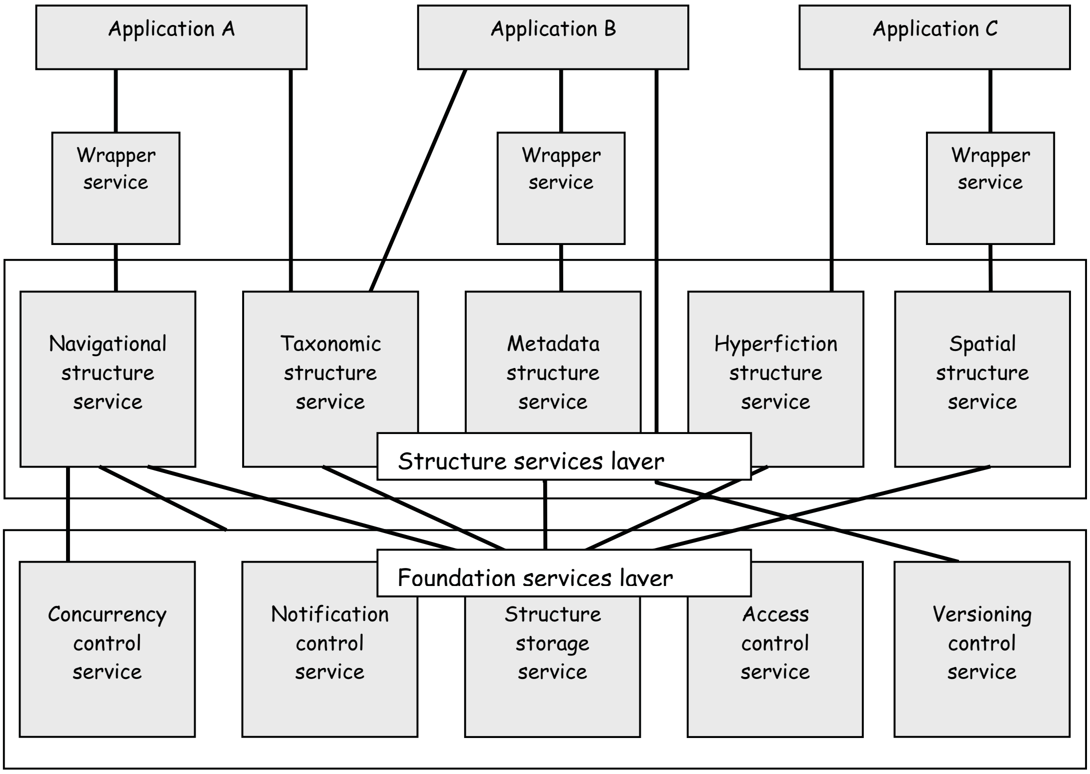


Figure 31. A conceptual overview of the multiple open service idea

6.3 Issues for Further Research

A challenging issue for further research is a time-ontology approach. With this approach, time ontology is an explicit artifact distinct from the resources related to an application. Maintainability can be regarded as one advantage of an ontology-driven approach. When the time ontology is created first, from which time rules could be generated and applied to appropriate resources, we can separate the ontology design from the resource space. The same time ontology can be overlaid on different pools of resources to provide different temporal views to users. With the ontology-driven approach possible existing time ontologies can be used instead of significant investment for creating them from the very beginning. Modelling general and domain specific temporal rules is an interesting research topic.

Applying an interval relation algebra presented by Hirsch (1996) to time-sensitive linking structures and developing (Extended) Backus-Naur Form (EBNF) like notation are two interesting directions in a more theoretical sense. The XML Linking Language (XLink) Version 1.1, which is the W3C Working Draft 28 April 2005, introduces non-normative Sample DTD, Sample XML Schema and Sample RELAX NG Grammar for XLink in the appendix part of the Working Draft (W3C 2005e). In XLink Version 1.0 no

Sample XML Schema and Sample RELAX NG Grammar were given. These could provide more formal representations for XLink and related extensions such as XLinkTime.

Semantic Web technologies will offer a challenging framework, in both a theoretical and practical sense, for researching and demonstrating metadata management of time-sensitive linking structures, management of linkbases and structures inside and between linkbases. On a more practical level, SVG offers interesting possibilities in ubiquitous environments for demonstrating 3D time-sensitive user interfaces.

Context-based linking and Topic Maps offer an interesting approach to knowledge integration in multidatabase environments. The advantage of this approach is knowledge integration at a metalevel, regardless of what the formats of the underlying databases are. With this approach we could contribute to solving knowledge sharing problems on the horizon, both in private and public sectors, caused by the mobility of labour and globalization of our world.

Bibliography

Akscyn, R. M., McCracken, D. L. and Yoder, E. A. 1988. KMS: A distributed hypermedia system for managing knowledge in organizations. Communications of the ACM, Vol. 31, No. 7, pp. 820 – 835.

Albers, M. J. 2003. Multidimensional analysis for custom content for multiple audiences. In: Proceedings of the 21st Annual International Conference on Documentation. New York, NY, USA: ACM Press. Pp. 1 – 5.

Allen, J. F. 1983. Maintaining knowledge about temporal intervals. Communications of the ACM, Vol. 26, No. 11, pp. 832 – 843.

Allen, J. F. 1984. Towards a general theory of action and time. Artificial Intelligence, Vol. 23, No. 2, pp. 123 – 154.

Allen, J. F. and Kautz, H. A. 1985. A model of naive temporal reasoning. In J. R. Hobbs and R. C. Moore (Eds.) Formal Theories of the Commonsense World. Toronto, Canada: Ablex Publishing Corp. Pp. 251 – 268.

Allen, J. F. 1991. Time and time again: The many ways to represent time. International Journal of Intelligent Systems, Vol. 6, No. 4, pp. 341 –355.

Anderson, K. M. 1997. Integrating open hypermedia systems with the World Wide Web. In: Bernstein, M., Carr, L. and Østerbye, K (Eds.) 8th ACM Conference on Hypertext, Hypertext '97, Southampton, UK, 6 – 11 April 1997. New York, NY, USA: ACM Press. Pp. 157 – 166.

Anderson, K. M. 2001. The extensibility mechanisms of the Chimera open hypermedia system. Journal of Network and Computer Applications, Vol. 24, No. 1, pp. 75 – 86.

Anderson, K. M., Taylor, R. N. and Whitehead, E. J. 1994. Chimera: Hypertext for heterogeneous software environments. In: Proceedings of the ACM European Conference on Hypermedia Technology (ECHT ´94), September, 18 – 23, Edinburgh, Scotland, UK. New York, NY, USA: ACM Press. Pp. 94 – 197.

Andrews, K., Kappe, F. and Maurer, H. 1995. Serving information on the Web with Hyper-G. Computer Networks and ISDN Systems, Vol. 27, No. 6, pp. 919 – 926.

Ashman, H. and Simpson, R.M. 1999. Computing Surveys´ Electronic Symposium on Hypertext and Hypermedia: Editorial. ACM Computing Surveys, Vol. 31, No. 4es.

AskOxford 2004. Search the Compact Oxford English Dictionary (referred 1.3.2005), <URI:http://www.askoxford.com/dictionaries/compact_oed/?view=uk>

Balasubramanian, V., Bieber, M. and Isakowitz, T. 2001. A case study in systematic hypermedia design. Information Management, Vol. 26, No. 4, pp. 295 – 320.

Barnett, M. 1998. Testing a digital library of technical manuals. IEEE Transactions on Professional Communication, Vol. 41, 2, pp. 116 – 122.

Beales, R., Cruickshank, D., De Roure, D., Gibbins, N., Juby, B., Michaelides, D. T. and Page, K. R. 2001. The Pipeline of enrichment: supporting link creation for continuous media. In: Proceedings of Openness, Structural Awareness, and Adaptivity: International Workshops. Berlin: Springer-Verlag. Pp. 47 – 58.

Bechhofer, S., Carr, L., Goble, C., Hall, W. 2001. Conceptual open hypermedia = the Semantic Web? (referred 1.3.2005), <URI: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS//Vol-40/Goble-et-al.pdf>.

Berk, E. and Devlin, J. (Eds.) 1991. Hypertext/hypermedia handbook. New York: Intertext Publications. 583 p.

Berners-Lee, T. 1999. Weaving the Web. The original design and ultimate destiny of the World Wide Web. New York: HarperSanFrancisco. 226 p.

Berners-Lee, T., Hendler, J. and Lassila, O. 2001. The Semantic Web. Scientific American, Vol. 284, No. 5, pp. 28 – 37.

Berztiss, A. T. 2002. Time in modelling. In: Kangassalo, H., Jaakkola, H., Kawaguchi, E. and Welzer, T. (Eds.). Frontiers in Artificial Intelligence and Applications, Vol. 94, Information Modelling and Knowledge Bases XIII. Amsterdam: IOS Press. Pp. 184 – 200.

Bieber, M., Vitali, F., Ashman, H., Balasubramanian, V. and Oinas-Kukkonen, H. 1997. Fourth generation hypermedia: some missing links for the World Wide Web. International Journal of Human-Computer Studies, Vol. 47, No. 1, pp. 31– 65.

Blakowski, G. and Steinmetz, R. 1996. A media synchronization survey: reference model, Specification and Case Studies, IEEE Journal on Selected Area in Communications, Vol. 14, No. 1, pp. 5 – 35.

Bouvin, N. O. 2000. Designing user interfaces for collaborative Web-based open hypermedia. In: ACM 2000 Hypertext. Proceedings of the 11th ACM Conference on Hypertext and Hypermedia, San Antonio, TX, USA, 30 May – 3 June 2000. New York, NY, USA: ACM. Pp. 230 – 231.

Bouvin, N. O. and Schade, R. 1999. Integrating temporal media and open hypermedia on the World-Wide Web. Computer Networks (Proceedings of the 8th International World Wide Web Conference, Toronto, Ont., Canada, 11 – 14 May 1999), Vol. 31. No. 11 – 16, pp. 1453 – 1465.

Boy, G. A. 1997. Active design documents. In: Proceedings of the Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques. New York, NY, USA: AMC Press. Pp. 31 – 36.

Brézillon, P. 2003. Focusing on context in human-centered computing. IEEE Intelligent Systems, Vol. 18, No. 3, pp. 62 – 66.

Brown, P. J. and Brown, H. 1995. Embedded or separate hypertext mark-up: is it an issue? Electronic Publishing, Vol. 8, No. 1, pp. 1 – 13.

Brusilovsky, P. 1996. Methods and techniques of adaptive hypermedia. User Modeling and User-Adapted Interaction, Vol. 6, No. 2 – 3, pp. 87 – 129.

Brusilovsky, P. and Maybury, M. T. 2002. From adaptive hypermedia to the adaptive Web. Communications of the ACM, Vol. 45, No. 5, pp. 30 – 33.

Brusilovsky, P., Stock, O. and Strapparava, C. (Eds.) 2002. Adaptive hypermedia and adaptive Web-based systems. Lecture Notes in Computer Science 1892, Berlin: Springer-Verlag. 422 p.

Bry, F. and Kraus, M. 2002. Advanced modelling and browsing of technical documents. In: Proceedings of the 2002 ACM Symposium on Applied Computing. New York, NY, USA: ACM Press. Pp. 520 – 524.

Buchanan, M. C. and Zellweger, P. T. 1992. Specifying temporal behavior in hypermedia documents. In: Proceedings of the ACM European Hypertext and Hypermedia Conference, Milano, Italy, November 30 – December 4, 1992. New York, NY, USA: ACM Press. Pp. 262 – 271.

Buchanan, M. C. and Zellweger, P. T. 1993. Automatic temporal layout mechanisms. In: Proceedings of the ACM Multimedia 93, Anaheim, California, August 1 – 6, 1993. New York, NY, USA: ACM Press. Pp. 341 – 350.

Bulterman, D. C. A., 2001. SMIL 2.0, Part 1: Overview, concepts, and structure. IEEE Multimedia, Vol. 8, No. 4, pp. 82 – 88.

Bulterman, D. C. A. 2002. SMIL 2.0 Part 2: Examples and comparisons. IEEE Multimedia, Vol. 9, No. 1, pp. 74 – 84.

Bush, V. 1945. As we may think. Atlantic Monthly, Vol. 176, No. 1, pp. 101 – 108.

Campbell, B. and Goodman, J. 1988. HAM: A general purpose hypertext abstract machine. Communications of the ACM, Vol. 31, No. 7, pp. 856 – 861.

Carr, L., De Roure, D., Hall, W. and Hill, G. 1995. The Distributed Link Service: A tool for publishers, authors and readers (referred 1.3.2005), <URI: http://www.w3.org/Conferences/WWW4/Papers/178/>.

Carr, L. A., Hall, W. and Hitchcock, S. 1998. Link services or link agents? In: Grønbæk, K., Mylonas, E. and Shipman, F. M. (Eds.) Proceedings of the 9th ACM Conference on Hypertext and Hypermedia, Pittsburgh, PA 20 – 24 June 1998, USA. New York, NY, USA: ACM Press. Pp. 113 – 122.

Christensen, B. G., Hansen, F. A. And Bouvin, N. O. 2003. Xspect: Bridging open hypermedia and Xlink. In: Proceedings of the WWW 2003, May 20 – 24, 2003, Budapest, Hungary. Pp. 490 – 499.

Ciancarini, P., Folli, F., Rossi, D. and Vitali, F. 2002. XLinkProxy: External linkbases with XLink. The New Review of Hypermedia and Multimedia, Vol. 8, 2002, pp. 27 – 37.

Comer, D. E 2001. Computer networks and Internets. With Internet applications. Upper Saddle River, New Jersey: Prentice-Hall, Inc. 603 p.

Conklin, E. J. 1987. Hypertext: an introduction and survey. IEEE Computer, Vol. 20, No. 9, pp. 17 – 41.

Cooke, P. and Williams, I. 1989. Design issues in large hypertext systems for technical documentation. In: McAleese, R. (Ed.). Hypertext: theory into practice. Osney Mead, Oxford: Blackwell Scientific Publications Ltd. Pp. 93 – 104.

Coulouris, G., Dollimore, J., Kindberg, T. 2001. Distributed systems. Concepts and design. Harlow, England: Pearson Education Inc. 772 p.

Courtiat, J.-P., Rust da Costa Carmo, L. F. and Cruz de Oliveira, R. 1996. A general-purpose multimedia synchronization mechanism based on causal relations. IEEE Journal on Selected Areas in Communications, Vol.14, No. 1, pp. 185 – 195.

Craig, J.S. 1992. Using a structured design analysis to simplify complex in-house computer manuals. IEEE Transactions on Professional Communication, Vol. 35, No. 1, pp. 7 – 12.

Cunliffe, D. 2000. Trailblazing: trends in hypermedia. The New Review of Hypermedia and Multimedia, Vol. 6, pp. 19 – 46.

Dave, P., Karadkar, U. P., Furuta, R., Francisco-Revilla, L., Shipman, F., Dash, S. and Dalal, Z. 2003. Browsing intricately interconnected paths. In: Proceedings of the 14th ACM Conference on Hypertext and Hypermedia, Nottingham, UK. New York, NY, USA: ACM Press. Pp. 95 – 103.

Davis. H. 1995. To embed or not to embed ... Communications of the ACM, Vol. 38, No. 6, pp. 108 – 109.

Davis, H. C., Millard, D. E., Reich, S., Bouvin, N., Grønbæk, K., Nürnberg, P. J., Sloth, L., Wiil, U. K. and Anderson, K. M. 1999. Interoperability between hypermedia systems: The Standardisation Work of the OHSWG. In: Proceedings of Hypertext '99, The 10th ACM Conference on Hypertext and Hypermedia, Darmstadt, February 21 – 25, 1999. New York, NY, USA: ACM Press. Pp. 201 – 202.

De Roure, D. and Hall, W. 1997. Distributed multimedia information systems. IEEE Multimedia, Vol. 4, No. 4, pp. 68 – 73.

De Roure, D.C., Walker, N.G. and Carr, L.A. 2000. Investigating link service infrastructures. In: Proceedings of the 11th ACM Conference on Hypertext and hypermedia, San Antonio, Texas, USA, New York, NY, USA: ACM Press. Pp. 67 – 76.

Deitel, H. M., Deitel, P. J. and Goldberg, A. B. 2003. Internet & World Wide Web. How to program. Upper Saddle River, New Jersey: Prentice-Hall, Inc. 1296 p.

DeRose, S. J. and Durand, D. G. 1994. Making hypermedia work. A user's guide to HyTime. Boston: Kluwer Academic Publishers. 384 p.

DeRose, S. J. and Durand, D. G. 1995. The TEI hypertext guidelines. Computing and the Humanities, Vol. 29, No. 3, pp. 181 – 190.

DocZilla 2003. DocZilla Home Page (referred 1.3.2005), <URI: http://www.doczilla.com/>.

El-Beltagy, S. R., Hall, W., De Roure, D. and Carr, L. 2001. Linking in context. In: Davis, H., Douglas, Y and Durand, D. G. (Eds.) Proceedings of the 12th ACM Conference on Hypertext and Hypermedia, Århus, Denmark. New York, NY, USA: ACM Press. Pp. 151 – 160.

Elmasri, R. and Navathe, S. B. 1994. Fundamentals of Database Systems. Redwood City, CA: The Benjamin/Cummings Publishing Company, Inc. 873 p.

Engelbart, D. C. 1963. A conceptual framework for the augmentation of man's intellect. In: Howerton, P. W. and Weeks, D. C. (Eds.). Vistas in Information Handling. Volume 1. The Augmentation of Man's Intellect by Machine. Washington, D.C., Spartan Books. Pp. 1 – 29.

Engelbart, D.C. 1984. Collaboration support provisions in Augment. (referred 1.3.2005), <URI: http://www.bootstrap.org/augdocs/oad-2221.htm>.

Fieldeing, R. T., Whitehead, E. J., Anderson, K. M., Bolcer, G., A., Oreizy, P. and Taylor, R. N. 1998. Web-based development of complex information products. Communications of the ACM, Vol. 41, No. 8, pp. 84 – 92.

Ford, N. 2004. Art of Java Web development. Greenwich, CT: Manning Publications Co. 593 p.

Fountain, A. M., Hall, W., Heath, I. and Davis, H. C. 1990. MICROCOSM: An open model for hypermedia with dynamic linking. In: Rizk, A., Streitz, N. and André, J. (Eds.) Hypertext: Concepts, Systems and Applications. Proceedings of the First European Conference on Hypertext, November 1990, France. INRIA. Pp. 298 – 311.

Furuta, R. and Na, J.-C. 2002. Applying caT`s programmable browsing semantics to specify World-Wide Web documents that reflect place, time, reader and community. In: Proceedings of the 2002 ACM Symposium on Document Engineering, McLean, Virginia, USA. New York, NY, USA: ACM Press. Pp. 10 – 17.

Furuta, R., Shipman, F. M., Marshall, C. C., Brenner, D. and Hsieh, H.-W. 1997. Hypertext paths and the World Wide Web: Experiences with Walden´s paths. In: Proceedings of the 8th ACM Conference on Hypertext, Southampton, United Kingdom. New York, NY, USA: ACM Press. Pp. 167 – 176.

Georganas, N. D, Steinmetz, R. and Nakagawa, T. 1996. Guest editorial synchronization issues in multimedia communications. IEEE Journal on Selected Areas in Communication, Vol. 14, No. 1 (referred 1.3.2005), <URI: http://www.argreenhouse.com/society/J-SAC/TOC/1996/jan96.html>.

Giumale, C. A. and Kahn, H. J. 1993. An information model of time. In: Proceedings of the 30th International Annual ACM IEEE Design Automation Conference, Dallas, Texas, USA. New York, NY, USA: ACM Press. Pp. 668 – 672.

Goldfarb, C. F. and Prescod, P. 1998. The XML Handbook™. Upper Saddle River, New Jersey: Prentice-Hall, Inc. 639 p.

Goldie, P. 1997. Using SGML to create complex interactive documents for electronic publishing. IEEE Transactions on Professional Communication, Vol. 40, No. 2, pp. 129 – 138.

Goose, S., Lewis, A. and Davis, H. 1997. OHRA: towards an open hypermedia reference architecture and a migration path for existing systems. JoDI - Journal of Digital Information, Vol. 1, No. 2 (referred 1.3.2005), <URI: http://journals.ecs.soton.ac.uk/jodi/Articles/v01/i02/Goose/>.

Gronbaek, K. and Wiil, U. K 1997. Towards a common reference architecture for open hypermedia. JoDI - Journal of Digital Information, Vol. 1, No. 2 (referred 1.3.2005), <URI: http://jodi.ecs.soton.ac.uk/Articles/v01/i02/Gronbak/>.

Gronbaek, K. and Trigg, R.H. 1994. Hypermedia system design applying the Dexter Model. Communications of the ACM**,** Vol. 37, No. 2, pp. 26 – 29.

Grønbæk, K. and Trigg, R. H. 1999. From Web to Workplace. Designing open hypermedia systems. Cambridge, Massachusetts, USA: The MIT Press. 386 p.

Grønbæk, K., Bouvin, N. O. and Sloth, L. 1997. Designing Dexter-based hypermedia services for the World Wide Web. In: Bernstein, M., Carr, L. and Østerbye, K. (Eds.) Proceedings of the 8th ACM Conference on Hypertext, Southampton, UK. New York, NY, USA: ACM Press. Pp. 146 – 156.

Grønbæk, K., Kyng, M. and Mogensen, P. 1993. CSCW challenges: Cooperative design in engineering projects. Communication of the ACM, Vol. 36, No. 4, pp. 67 – 77.

Gronbaek, K., Sloth, L. and Bouvin, N. O. 2000. Open hypermedia as user controlled meta data for the Web. Computer Networks (Proceedings of the 9th International World Wide Web Conference. Amsterdam, Netherlands, 15 – 19 May 2000), Vol. 33, No. 1– 6, pp. 553 – 566.

Grønbæk, K., Sloth, L. and Orbaek, P. 1999. Webvise: browser and proxy support for open hypermedia structuring mechanisms on the World-Wide Web. Computer Networks (Proceedings of the 8th International World Wide Web Conference. Toronto, Ont., Canada, 11 – 14 May 1999), Vol. 31, No. 11 – 16, pp. 1331 – 1345.

Halasz, F. 1988. Reflections on Notecards: Seven issues for the next generation of hypermedia systems. Communications of the ACM, Vol. 31, No. 7, pp. 836 – 852.

Halasz, F. 1991. "Seven Issues" Revisited. (referred 1.3.2005), <URI: http://www2.parc.com/spl/projects/halasz-keynote/>.

Halasz, F. and Schwartz, M. 1994. The Dexter hypertext reference model. Communications of the ACM, Vol. 37, No. 2, pp. 30 – 39.

Hall, W. 2000. The button strikes back. The New Review of Hypermedia and Multimedia, Vol. 6, pp. 5 – 17.

Hall, W. and Davis, H. 1994. Hypermedia link service and their application to multimedia information management. Information and Software Technology, Vol. 36, No. 4, pp. 197 – 202.

Hall, W., Davis, H., Hutchings, G., Hall, W., Davis, H. C. and Hutchings, G. A. 1996. Rethinking hypermedia: The Microcosm approach. Boston, MA, USA: Kluwer Academic Press. 195 p.

Hall, W., Heath, I., Hill, G. J., Davis, H. C., Wilkins, R. J. 1992. The design and implementation of an open hypermedia system (referred 1.3.2005), <URI: http://www.mmrg.ecs.soton.ac.uk/publications/archive/hall1992/>.

Hameri, A-P. 1997. Project management in a long-term and global one-of-a-kind project. International Journal of Project Management, Vol. 15, No. 3, pp. 151 – 157.

Hameri, A.-P. and Heikkilä, J. 2002. Improving efficiency: Time-critical interfacing of project tasks. International Journal of Project Management, Vol. 20, No. 2, pp. 143 – 153.

Hameri, A.-P. and Nihtilä, J. 1997. Distributed new product development project based on Internet and World-Wide Web: A case study. The Journal of Product Innovation Management, Vol. 14, No. 2, pp. 77 – 87.

Hameri, A.-P. and Nihtilä, J. 1998. Data-based learning in product development. Scandinavian Journal of Management, Vol. 14, No. 3, pp. 223 – 238.

Hardman, L. 2000. Modelling and authoring hypermedia documents. (referred 1.3.2005), <URI: http://homepages.cwi.nl/~lynda/thesis/>.

Hardman, L. and Bulterman, D., C., A. 1997. Document model issues for hypermedia. In: Grosky, W. I., Jain, R. and Mehrotra, R. (Eds.) The Handbook of Multimedia Information Management. Upper Saddle River, NJ: Prentice Hall PTR. Pp. 39 – 68.

Hardman, L., Bulterman, D. C. A. and van Rossum, G. 1993a. Links in hypermedia: The requirement for context. In: Proceedings of the 5th ACM Conference on Hypertext and Hypermedia, Seattle, Washington, USA, November 14 – 18, 1993. New York: ACM Press. Pp. 183 – 191.

Hardman, L., Bulterman, D. C. A. and van Rossum, G. 1993b. Structured multimedia authoring. In: Proceedings of the ACM Multimedia ´93 Conference, Anaheim, California, USA, August 1 – 6, 1993. New York: ACM Press. Pp. 283 –289.

Hardman, L., Bulterman, D. C. A. and van Rossum, G. 1994. The Amsterdam hypermedia model: Adding time and context to the Dexter model. Communications of the ACM, Vol. 37, No. 2, pp. 50 – 62.

Hardman, L., Schmitz, P., van Ossenbruggen, J., ten Kate, Warner and Rutledge, L. 2000. The link vs. the event: activating and deactivating elements in time-based hypermedia. The New Review of Hypermedia and Multimedia, Vol. 6, pp. 89 – 109.

Hardman, L., Van Ossenbruggen, J., Mullender, S. K., Rutledge, L., Bulterman, D. C.A. 1999. Do you have the time? Composition and linking in time-based hypermedia. In. Proceedings of the 10th ACM Conference on Hypertext and Hypermedia, February 21 – 25, 1999, Darmstadt, Germany. New York, NY, USA: ACM Press. Pp. 189 – 196.

Hirashima, T., Matsuda, N., Nomoto, T., Toyoda, J. 1997. Context-sensitive filtering for browsing in hypertext. In: Proceedings of the 3rd International Conference on Intelligent User Interfaces, San Francisco, California, USA. New York, NY, USA: ACM Press. Pp. 119 – 126.

Hirsch, R. 1996. Relation algebras of intervals. Artificial Intelligence, Vol. 83, No. 2, pp. 267 – 295.

ISO 8601. 2000. Data elements and interchange formats - Information interchange - Representation of dates and times. Geneva: International Organization for Standardization. 29 p.

ISO 8879. 1986. Information processing – Text and office systems – Standard generalized markup language (SGML). Geneva: International Organization for Standardization. 155 p.

ISO/IEC 10744. 1997. Information processing - Hypermedia/Time-based Structuring Language (HyTime) - 2d edition (referred 1.3.2005), <URI: http://www.ornl.gov/sgml/wg8/docs/n1920//>.

ISO/IEC 13250. 2000. Information technology - SGML Applications - Topic Maps. Geneva: International Organization for Standardization. 39 s.

ISO/IEC 13522-1. 1997. Information technology - Coding of multimedia and hypermedia information - Part 1: MHEG object representation - Base notation (ASN.1). Geneva: International Organization for Standardization. 357 p.

ISO/IEC 13522-8. 2001 Information technology - Coding of multimedia and hypermedia information - Part 8: XML notation for ISO/IEC 13522-5 Geneva: International Organization for Standardization. 20 p.

J2EE 2004. Java Servlet Technology. (referred 1.3.2005), <URI: http://java.sun.com/products/servlet/index.jsp>.

Jacobson, I., Booch, G. and Rumbaugh, J. 1999. The unified software development process. Boston: Addison-Wesley. 463 p.

Jourdan, M. 2001. A formal semantics of SMIL: a Web standard to describe multimedia documents. Computer Standards & Interfaces, Vol. 23, No. 5, pp. 439 – 455.

Jühne, J., Jensen, A.T. and Gronbaek, K. 1998. Ariadne: A Java-based guided tour system for the World Wide Web. Computer Networks and ISDN Systems, Vol. 30, No. 1 – 7, pp. 131 – 139.

Järvinen, P. 2001. On research methods. Tampere, Finland: Opinpajan kirja. 189 p.

Järvinen, P. 2004. On research methods. Tampere, Finland: Opinpajan kirja. 204 p.

Karousos, N., Pandis, I., Reich, S. and Tzagarakis, M. 2003. Offering open hypermedia services to the WWW: A step-by-step approach for developers (referred 1.3.2005), <URI: http://www2003.org/cdrom/papers/refereed/p531/p531-karousos.htm>:

Ketamo, H. 2002. User and platform adaptation in Web-based learning environments. Tampere University of Technology Publications 381, Tampere, Tampere University of Technology. 142 p.

Koivunen, M.–R. and Miller, E. 2002. W3C Semantic Web activity. In: Hyvönen, E. (ed.) Semantic Web Kick-Off in Finland. Visions, Technologies, Research and Applications, November, 2, 2001, Helsinki, Finland. HIIT Publications 2002-01. Pp. 27 – 43.

Komathy, K., Vivekanandan, P. and Ramachandran, V. 2003. Secure SOAP-based Web Services. Computer Systems: Science and Engineering, Vol. 18, No. 1, pp. 27 – 33.

Kronodoc 2004. Kronodoc Home Page (referred 1.3.2005), <URI: http://www.kronodoc.fi/ >.

Lassila, O., van Harmelen, F., Horrocks, I., Hendler, J. and McGuinness, D.L. 2000. The Semantic Web and its languages. IEEE Intelligent Systems, Vol. 15, No. 6, pp. 67 – 73.

Lemahieu, W. 2002. Context-based navigation in the Web by means of dynamically generated guided tours. Computer Network, Vol. 39, No. 3, pp. 311 – 328

Liebowitz, J. and Megbolugbe, I. 2003. A set of frameworks to aid the project manager in conceptualizing and implementing knowledge management initiatives. International Journal of Project Management, Vol. 21, No. 3, pp. 189 – 198.

March, S. T. and Smith, G. F. 1995. Design and natural science research on information technology. Decision Support Systems, Vol. 15, No. 4, pp. 251 – 266.

Martínez, J.M. 2002. MPEG-7 Overview of MPEG-7 description tools, Part 2. IEEE Multimedia, Vol. 9, No. 3, pp. 83 – 93

Martínez, J. M., Koenen, R. and Pereira, F. 2002. MPEG-7 The generic multimedia content description standard, Part 1. IEEE Multimedia, Vol. 9, No. 2, pp. 78 – 87.

Maurer, H. 1996. Hyper-G now HyperWave™. The next generation Web solution. Essex, UK: Addison Wesley Longman. 635 p.

McGuire, M. E., Henderson, G. D., Moran, J. M., Center, L. A. and Ely, L. 1998. Combining training and customer documentation into modular, reusable information. In: Proceedings of the 16th Annual International Conference on Computer Documentation. New York, NY, USA: ACM Press. Pp. 18 – 22.

Meyrowitz, N. 1989. The missing link: Why we´re all doing hypertext wrong. In: The Society of Text: Hypertext, Hypermedia, and the Social Construction of Information. Cambridge, MA, USA: MIT Press. Pp. 107 – 114.

Michaelides, D. T., Millard, D. E., Weal, M. J., De Roure, D. 2001. Auld Leaky: A contextual open hypermedia link server, Revised Papers from the International Workshops OHS-7, SC-3, and AH-3 on Hypermedia: Openness, Structural Awareness, and Adaptivity, August 14 – 18, 2001, Aarhus, Denmark. Pp. 59 – 70.

Millard, D., Davis, H., Weal, M. Aben, K. and De Bra, P. 2003. AHA! Meet Auld Linky: Integrating designed and free-formed hypertext systems. In: Proceedings of the 14th ACM Conference on Hypertext and Hypermedia, August 26 – 30, 2003, Nottingham, UK. New York: ACM Press. Pp. 161 – 169.

Millard, D.E., Moreau, L., Davis, H.C., Reich, S. 2000. FOHM: a fundamental open hypertext model for investigating interoperability between hypertext domains. In: Proceedings of the 11th ACM Conference on Hypertext and Hypermedia, May 30 – June 4, 2000, San Antonio, Texas, USA. New York: ACM Press. Pp. 93 – 102.

Mozilla 2005. Mozilla Homepage (referred 1.3.2005), <URI: http://www.mozilla.org/>.

Muchaluat-Saade, D. C. and Soares, L. F. G. 2002. XConnector and XTemplate: improving the expressiveness and reuse in Web authoring languages. The New Review of Hypermedia and Multimedia, Vol. 8, 2002, pp. 139 – 169.

Multisilta, J. 1996. Hypermedia learning environment for mathematics. Tampere University of Technology Publications 183, Tampere, Tampere University of Technology. 221 p.

Na, J-C. and Futura, R. 2000. Context-aware hypermedia in a dynamically-changing environment, supported by a high-level Petri net. In: Proceedings of the 11th ACM on Hypertext and Hypermedia, San Antonio, Texas, USA. New York: ACM Press. Pp. 222 – 223.

Nack, F. and Lindsay, A. T. 1999a. Everything you wanted to know about MPEG-7: Part 1. IEEE Multimedia, Vol. 6, No. 3, pp. 65 – 77.

Nack, F. and Lindsay, A. T. 1999b. Everything you wanted to know about MPEG-7: Part 2. IEEE Multimedia, Vol. 6, No. 4, pp. 64 – 73.

Nanard, J. and Nanard, M. 1995. Hypertext design environments and the hypertext design process. Communications of the ACM, Vol. 38, No. 8, pp. 49 – 56.

Nanard, J., Nanard, M. and King, P. 2000. Media construction formalism specifying abstractions for multimedia scenario design. The New Review of Hypermedia and Multimedia, Vol. 6, pp. 47 – 87.

Nelson, T. H. 1987. Literary Machines. Edition 87.1, San Antonio, TX, Project Xanadu. 263 p.

Nelson, T.H. 1995. The heart of connection: Hypermedia unifiedby transclusion. Communication of the ACM, Vol. 38, No. 8, pp. 31 – 33.

Nelson, T.H. 1999. Xanalogical structure, needed now more than ever: Parallel documents, deep links to content, deep versioning, and deep re-use. ACM Computing Surveys, Vol. 31, No. 4es, Article No. 33.

Newcomb, S. R., Kipp, N. A. and Newcomb, V. T. 1991. The “HyTime”. Hypermedia/Time-based document structuring Language. Communications of the ACM, Vol. 34, No. 11, pp. 67 – 83.

Nielsen, J. 1995. Multimedia and hypertext. Massachusetts, MA, Cambridge: AP Professional. 480 p.

Norton, D. W. 2000. Technical communication as business strategy: How changes in discursive patterns affect the value of technical communication in cross-functional team settings. IEEE Transactions on Professional Communication, Vol. 43, No. 1, pp. 77 – 89.

Nunamaker, Jr. J. F., Chen, M. and Purdin, T. D. M. 1991. Systems development in information systems research. Journal of Management Information Systems, Vol. 7, No. 3, pp. 89 – 106.

Nürnberg, P. J. and Ashman, H. 1999. What was the question? Reconciling open hypermedia and World Wide Web research. In: Tochtermann, K., Westbomke, J., Wiil, U. K. and Leggett, J. J. (Eds.) Hypertext '99. Returning to our Diverse Roots. The 10th ACM Conference on Hypertext and Hypermedia, Darmstadt, Germany, 21 – 25 Feb 1999. New York, NY, USA: ACM. Pp. 83 – 90.

Nürnberg, P. J. and Leggett, J. J. 1997. A vision for open hypermedia systems. Journal of Digital Information, Vol. 1, No. 2 (referred 1.3.2005), <URI: http://jodi.ecs.soton.ac.uk/Articles/v01/i02/Nurnberg/>

Nürnberg, P. J., Leggett, J. J. and Schnase, J. L. 1996. Hypermedia operating systems: a new paradigm for computing. In: Proceedings of the 7th ACM Conference on Hypertext and Hypermedia, Washington DC 16 – 20 March 1996, USA. New York, NY, USA: ACM Press. Pp. 194 – 202.

Nürnberg, P. J., Leggett, J. J. and Schneider, E. R. 1997. As we should have thought. In: Proceedings of the 8th ACM Conference on Hypertext and Hypermedia, Southampton, United Kingdom, April 6 – 11, 1997. New York, NY, USA: ACM Press. Pp. 96 – 101.

Nürnberg, P. J., Leggett, J. J. and Wiil, U. K. 1998. An agenda for open hypermedia research. In: Grønbæk, K., Mylonas, E. and Shipman, F. M. (Eds.) Proceedings of the 9th ACM Conference on Hypertext and Hypermedia, Pittsburgh, PA 20 – 24 June 1998, USA. New York, NY, USA: ACM Press. Pp. 198 – 206.

Nürnberg, P.J. and Schraefel, M. M. C. 2003 Relationships among structural computing and other fields. Journal of Network and Computer Applications**,** Vol. 26, No. 1, pp. 11 – 26

OASIS 2005. OASIS Cover pages: XML Linking Language (referred 1.3.2005), <URI: http://www.oasis-open.org/cover/xll.html/>.

Oinas-Kukkonen, H. 1998. What is inside a link? Communications of the ACM, Vol. 41, No. 7, p. 98.

Oinas-Kukkonen, H. 2000. Representing metaknowledge through rich links. In: Kawaguchi, E., Kangassalo, H., Jaakkola, H. and Hamid, I. A. (Eds.). Frontiers in Artificial Intelligence and Applications, Vol. 61, Information Modelling and Knowledge Bases XIV. Amsterdam: IOS Press. Pp. 275 – 282.

Page, K., Juby, B., Beales, R. and Roure, D. D. 2001. Continuous metadata. In: Proceedings of the 2nd Annual PostGraduate Symposium on The Convergence of Telecommunications, Networking & Broadcasting, pp. 264 – 269.

Park, J. and Hunting, S. 2003. XML Topic Maps. Creating and using Topic Maps for the Web. Boston: Addison-Wesley. 605 p.

Pérez-Luque, M. J. and Little, T. D. C. 1996. A temporal reference framework for multimedia synchronization. IEEE Journal on Selected Areas in Communications, Vol. 14, No. 1, pp. 36 – 51.

Pillai, A. S., Joshi, A. and Rao, K. S. 2002. Performance measurement of R&D projects in a multi-project, concurrent engineering environment. International Journal of Project Management, Vol. 20, No. 2, pp. 165 – 177.

Pitkow, J. E. and Jones, R. K. 1996. Supporting the Web: a distributed hyperlink database system. Computer Networks and ISDN Systems, Vol. 28, No. 7 – 11, pp. 981 – 991.

Price, J. 1997. Introduction: Special issue on structuring complex information for electronic publication. IEEE Transactions of Professional Communication, Vol. 40, No. 2, pp. 69 – 77.

Päivärinta, T. 2001. A genre-based approach to developing electronic document management in the organization. Jyväskylä Studies in Computing 11. Jyväskylä, Finland: Jyväskylä University Printing House 91 p. + included articles.

Rauch, T., Leone, P. and Gillihan, D. 1997. Enabling the book metaphor for the World Wide Web: Disseminating on-line information as dynamic Web documents. IEEE Transactions on Professional Communication, Vol. 40, No. 2, pp. 11 – 128.

Reich, S., Wiil, U. K., Nurnberg, P. J., Davis, H. C., Gronbaek, K., Anderson, K.M., Millard, D.E. and Haake, J. M. 1999. Addressing interoperability in open hypermedia: the design of the open hypermedia protocol. New Review of Hypermedia and Multimedia, Vol. 5, pp. 207 – 248.

Rizk, A. and Sutcliffe, D. 1997. Distributed link service in the Aquarelle project. In: Bernstein, M., Carr, L. and Østerbye, K. (Eds.) Proceedings of the 8th ACM Conference on Hypertext, Southampton, UK. New York, NY, USA: ACM Press. Pp. 208 – 227.

Schneiderman, B. 1987. User Interface Design for the Hyperties electronic encyclopedia. In: Proceedings of the ACM Conference on Hypertext**,** Chapel Hill, North Carolina, USA. New York, NY, USA: ACM Press. Pp. 189 – 194.

Schreiber, F. A. 1994. Is time a real time? An overview of time ontology in informatics. In: Halang, W. A. and Stoyenko, A. D. (Eds.) Real time computing. Proceedings of the NATO Advanced Study Institute, Sint Maarten, Dutch Antilles, 5 – 17 October, 1992. Berlin, Germany: Springer-Verlag. Pp. 283 – 307.

Sénac, P., De Saqui-Sannes, P. and Willrich, R. 1995. Hierarchical time stream Petri Net: A model for hypermedia systems. In: Proceedings of the 16th International Conference on Application and Theory of Petri Nets, June 26 – 30, Turin, Italy. Pp. 451 – 470.

Shipman III, F. M. and Hsieh, H. 2000. Navigable history: a reader's view of writer's time. The New Review of Hypermedia and Multimedia, Vol. 6, pp. 147 – 167.

Soares, L. F. G., Rodrigues, R. F. and Muchaluat-Saade, D. C. 2000. Modelling, authoring and formatting hypermedia documents in the HyperProp system. Multimedia Systems, Vol. 8, No. 2, pp. 118 – 134.

Staff, C. D. 2002. The hypercontext framework for adaptive hypertext. In: Proceedings of the 13th ACM Conference on Hypertext and Hypermedia, College Park, Maryland, USA. New York, NY, USA: Pp. 11 – 20.

Stotts, P. D. and Futura, R. 1989. Petri-net-based hypertext: document structure with browsing semantics. ACM Transactions on Information Systems, Vol. 7, No. 1, pp. 3 – 29.

Stotts, P. D. and Futura, R. 1990. Hierarchy, composition, scripting languages and translators for structured hypertext. In: Streitz, N., Rizk, A. and André, J. (Eds.). Hypertext: Concepts, systems and applications. Proceedings of the First European Conference on Hypertext, Versailles, France, November 1990. New York: Cambridge University Press. Pp. 180 – 193.

Stotts, P.D. and Furuta, R. 1991. Dynamic adaptation of hypertext structure. In: Proceedings of the 3rd Annual ACM Conference on Hypertext, San Antonio, Texas, USA. New York, NY, USA: ACM Press. Pp. 219 – 231.

Sumiya, K., Noda, R., and Tanaka, K. 2001. A temporal link mechanism for hypermedia broadcasting. Systems. Computers in Japan, Vol. 32, No. 14, pp. 54 – 66.

Takahashi, K. 1998. Metalevel links: More power to your links. Communications of the ACM, Vol. 41, No. 7, pp. 103 – 105.

Tong, A. K. Y. 2001. Linking and timing information presentation in multimedia educational systems. Journal of Educational Multimedia and Hypermedia, Vol. 10, No. 2, pp. 185 – 203.

Van Aken, J. E. 2004. Management research based on the paradigm of design sciences: The quest for field-tested and grounded technological rules. Journal of Management Studies, Vol. 41, No. 2, pp. 219 – 246.

Van Benthem, J. 1991. The logic of time. A model-theoretic investigation into the varieties of temporal ontology and temporal discourse. Dordrecht, The Netherlands: Kluwer Academic Publishers. 300 p.

Van Dam, A. 1988 Hypertext '87 Keynote Address, Special issue: Hypertext. Communications of the ACM, Vol. 31, No. 7, pp. 887 – 895.

Van Ossenbruggen, J., Hardman, L. and Rutledge, L. 2002. Hypermedia and the Semantic Web: A research agenda. Journal of Digital Information, Vol. 3, No. 1 (referred 2.3.2005), <URI: http://jodi.ecs.soton.ac.uk/Articles/v03/i01/VanOssenbruggen/>.

Vazirgiannis, M. 1999. Interactive multimedia documents. Modelling, authoring, and implementation experiences. Lecture Notes in Computer Science 1564. Berlin: Springer-Verlag. 161 p.

Vitali, F. and Bieber, M. 1999. Hypermedia on the Web: What will it take. ACM Computing Surveys, Vol. 31, No. 4es, Article No. 31.

Vivekanandan, S. M. and De Roure, D. 2002. Workflow Description for Open Hypermedia Systems. In: Proceedings of 2002 Workshop on Open Hypermedia Systems (OHS2002) Maryland, USA.

W3C 1998a. The World Wide Web Consortium: Date and Time Formats (referred 19.1.2005), <URL: http://www.w3.org/TR/NOTE-datetime/>.

W3C 1998b. The World Wide Web Consortium: Timed Interactive Multimedia Extensions for HTML (HTML+TIME), Extending SMIL into the Web Browser (referred 8.9.2005), <URL: http://www.w3.org/TR/NOTE-HTMLplusTIME/>.

W3C 1999. The World Wide Web Consortium: XML Path Language (XPath) Version 1.0 W3C Recommendation 16 November 1999 (referred 5.2.22002), <URI: http://www.w3.org/TR/xpath/>.

W3C 2000. The World Wide Web Consortium: Harvesting RDF Statements from XLinks W3C Note 29 September 2000 (referred 2.3.2005), <URI: http://www.w3.org/TR/xlink2rdf/>.

W3C 2001a. The World Wide Web Consortium: XML Base W3C Recommendation 27 June 2001 (referred 11.6.2004), <URI: http://www.w3.org/TR/xmlbase/>.

W3C 2001b. The World Wide Web Consortium: XML Linking Language (XLink) Version 1.0 W3C Recommendation 27 June 2001 (referred 2.3.2005), <URI: http://www.w3.org/TR/xlink/>.

W3C 2002. The World Wide Web Consortium: The World Wide Web Consortium: HLink: Link recognition for the XHTML W3C Working Draft 13 September 2002 (referred 2.3.2005), <URI: http://www.w3.org/TR/2002/WD-hlink-20020913/>.

W3C 2003a. The World Wide Web Consortium: SOAP 1.2 Implementation Summary (referred 2.3.2005), <URI: http://www.w3.org/2000/xp/Group/2/03/soap1.2implementation.html>.

W3C 2003b. The World Wide Web Consortium: XPointer xmlns() Scheme W3C Recommendation 25 March 2003 (referred 2.3.2005), <URI: http://www.w3.org/TR/xptr-xmlns/>.

W3C 2003c. The World Wide Web Consortium XPointer Framework W3C Recommendation 25 March 2003 (referred 2.3.2005), <URI: http://www.w3.org/TR/xptr-framework/>.

W3C 2003d. The World Wide Web Consortium: XPointer Element() Scheme. W3C Recommendation 25 March 2003 (referred 2.3.2005), <URI: http://www.w3.org/TR/xptr-element/>.

W3C 2004a. The World Wide Web Consortium: Extensible Markup Language (XML), 1.0 (Third Edition) W3C Recommendation 4th February 20 (referred 2.3.2005), <URI: http://www.w3.org/XML/>.

W3C 2004b. The World Wide Web Consortium: HyperText Markup Language (HTML) Home Page (referred 2.3.2005), <URI: http://www.w3.org/MarkUp/>.

W3C 2004c. The World Wide Web Consortium: RDF Vocabulary Description Language 1.0: RDF Schema W3C Recommendation 10 February 2004 (referred 2.3.2005), <URI: http://www.w3.org/TR/rdf-schema/>.

W3C 2004d. The World Wide Web Consortium: Semantic Web (referred 2.3.2005), <URI: http://www.w3.org/2001/sw/>.

W3C 2004e. The World Wide Web Consortium: The Resource Description Framework (RDF) (referred 2.3.2005), <URI: http://www.w3.org/RDF/>.

W3C 2004f. The World Wide Web Consortium: Web-Ontology (WebOnt) Working Group (referred 2.3.2005), <URI: http://www.w3.org/2001/sw/WebOnt/>.

W3C 2004g. The World Wide Web Consortium: Web Services Description Language (WSDL) Version 2.0 Part 0: Primer W3C Working Draft 21 December 2004 (referred 2.3.2005), <URI: http://www.w3.org/TR/2004/WD-wsdl20-primer-20041221/>.

W3C 2004h. The World Wide Web Consortium: Web Services Glossary, W3C Working Group Note 11 February 2004 (referred 2.3.2005), <URI: http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/>.

W3C 2005a. The World Wide Web Consortium: Scalable Vector Graphics (SVG) (refereed 2.3.2005), <URL: http://www.w3.org/Graphics/SVG/>.

W3C 2005b. The World Wide Web Consortium: The Annotea Project (referred 2.3.2005), <URI: http://www.w3.org/2001/Annotea/>.

W3C 2005c. The World Wide Web Consortium: The Synchronized Multimedia Integration Language (SMIL 2.0) W3C Recommendation 7 January 2005 (referred 6.6.2005), <URI: http://www.w3.org/TR/2005/REC-SMIL2-20050107/>.

W3C 2005d. The World Wide Web Consortium: The Synchronized Multimedia Integration Language (SMIL 2.1) W3C Candidate Recommendation 13 May 2005 (referred 6.6.2005), <URI: http://www.w3.org/TR/2005/CR-SMIL2-20050513/>.

W3C 2005e. The World Wide Web Consortium: XML Linking Language (XLink) Version 1.1 W3C Working Draft 28 April 2005 (referred 4.5.2005), <URI: http://www.w3.org/TR/2005/WD-xlink11-20050428>.

Waite, B. 2001. Hypermedia systems in the new millennium. ACM Journal of Computer Documentation, Vol. 25, No. 3, pp. 69 – 114.

Wang, H.-K.and Wu, J.-L. C. 1995. Interactive hypermedia applications: Model and its implementation. Software — Practice and Experience, Vol. 25, No. 9, pp. 1045 – 1063.

Wang, W. and Rada, R. 1998. Structured hypertext with domain semantics. ACM Transactions on Information Systems, Vol. 16, No. 4, pp. 372 – 412.

Wang, W., Haake, J. M., Rubart, J. and Tietze, D. A. 2001. Hypermedia-based support for cooperative learning of process knowledge. Journal of Network and Computer Applications, Vol. 23, No. 4, pp. 357 – 379.

Weal, M.J., Millard, D.E., Michaelides, D.T., De Roure, D.C. 2001. Building narrative structures using context based linking. In: Proceedings of the 12th ACM Conference on Hypertext and Hypermedia, Århus, Denmark. New York, NY, USA: ACM Press. Pp. 37 – 38.

Wei, K-K., Tan, B. C. Y., Sia, C-L and Raman, K. S. 1996. Hypertext: A new approach to construct group support systems. International Journal of Information Management, Vol. 16, No. 3, pp. 163 – 181.

Weinreich, H., Obendorf, H., Lamersdorf, W. 2001. The Look of the Link – Concepts for the User Interface of Extended Hyperlinks. In: Proceedings of the 12th ACM Conference on Hypertext and Hypermedia, Aarhus, Denmark. New York, NY, USA: ACM Press. Pp. 19 – 28.

Whitehead, J. and Wiggins, M. 1998. WEBDAV: IETF standard for collaborative authoring on the Web. IEEE Internet Computing, September/October, Vol. 2, No. 5, pp. 34 – 40.

Wiil, U. K. and Leggett, J. J. 1992. Hyperform: using extensibility to develop dynamic, open and distributed hypertext systems. In: Proceedings of the ACM Conference on Hypertext, Milano, November 30 – December 4, 1992 Italy. New York, NY, USA: ACM Press. Pp. 251 – 261.

Wiil, U. K. and Leggett, J. J. 1996. The HyperDisco approach to open hypermedia systems. In: Proceedings of the 7th ACM Conference on Hypertext and Hypermedia, Washington DC 16 – 20 March 1996, USA. New York, NY, USA: ACM Press. Pp. 140 – 148.

Wiil, U. K. and Leggett, J. J. 1997a. Hyperform: a hypermedia system development environment. ACM Transactions on Information Systems, Vol. 15, No. 1, pp. 1 – 31.

Wiil, U. K. and Leggett, J. J. 1997b. Workspaces: the HyperDisco approach to Internet distribution. In: Bernstein, M., Carr, L. and Østerbye, K. (Eds.) Proceedings of the 8th ACM Conference on Hypertext, Southampton, UK. New York, NY, USA: ACM Press. Pp. 13 – 23.

Wiil, U. K., Hicks, D. L. and Nürnberg, P. J. 2001. Multiple open services: a new approach to service provision in open hypermedia systems. In: Davis, H., Douglas, Y and Durand, D. G. (Eds.) Proceedings of the 12th ACM Conference on Hypertext and Hypermedia, Århus, Denmark. New York, NY, USA: ACM Press. Pp. 83 – 92.

XTM TopicMaps.Org. 2001. XML Topic Maps (XTM) 1.0. TopicMaps.Org Specification (referred 2.3.2005), <URI: http://www.topicmaps.org/xtm/1.0/>.

XUL 2005. XULPlanet (referred 2.3.2005), <URI: http://www.xulplanet.com/>.

Yankelovich, N., Meyrowitz, N. and van Dam, A. 1985. Reading and writing the electronic book. IEEE Computer, Vol. 18, No. 10, pp. 15 – 30.

Yankelovich, N., Haan, B. J., Meyrowitz, N. K. and Drucker, S. M. 1988. Intermedia: the concept and construction of a seamless information environment. IEEE Computer, Vol. 21, No. 1, pp. 81 – 96.

Yoo, J. and Bieber, M. 2001 A systematic relationship analysis for modelling information domains. In: Rossi, M. and Siau, K. (Eds.) Information modelling in the new millennium. Hershey, USA: Idea Group Publishing. Pp. 485 – 501.

Zhou, Q. and Fikes, R. 2002. A reusable time ontology (referred 18.1.2003), <URL: http://www. ksl.stanford.edu/KSL_Abstracts/KSL-00-01.html/>.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O. Box 527
FIN-33101 Tampere, Finland